BIRP: Bank Intra Routing Protocol

Ramzi A. Haraty, Rabie Sadek, and Ilhyun Lee

***Abstract*----SWIFT is a financial institution, which provides an interface for banks in order to communicate and transfer money worldwide. These banks, together with their branches incur significant expenses including security infrastructure, SAA package license, yearly maintenance, and message costs. Banks always look for a good investment and the swift routing protocol does not deal with intra bank message processing as a separate case. If a message is sent for either a branch or another bank, it must pass through SWIFT before it continues to the correspondent. In this work, we provide a new protocol - Bank Intra Routing Protocol (BIRP) - that intra routes all messages exchanged between the main branch and all its branches.**

***Index Terms*----Bank SWIFT and intra routing protocols.**

I. INTRODUCTION

SWIFT [1] is a financial organization that provides all banks with a dynamic graphical interface known as SWIFT Alliance Access (SAA) in order to communicate and transfer money all over the world. SAA is a financial message switch used to interface to multiple networks in order to transfer money in the most secure and consistent way. All banks are directly connected to SWIFT via SwiftNet secure links so that they can communicate with each other. Every message sent by the sender must go to SWIFT and from there it will be forwarded to the correspondent. Swift messages pass through many queues before they are moved to SWIFT. First, they are created in the message creation queues, and then are routed to both the verification and the authorization queues where they can be verified and authorized before continuing their tour to SWIFT. The main problem of SWIFT is that every message sent to a branch must pass through SWIFT before going to the receiver and this leads to additional costs to the bank. Besides that, if the connection is down, then all branch messages will fail to be received and must wait until the connection is up again. This delay increases as network traffic rises.

Institutions that have many branches in several countries have lots of expenses including security infrastructure, SAA package license, yearly maintenance, and message costs. Banks always look for a good investment and that can be achieved only if: message costs are saved, messages are received with less delay, and connectivity is trusted. The SWIFT routing protocol does not deal with intra messages exchanged between branches as a separate case, so all messages will have the same flow regardless of whether the receiver is a branch or any other bank. That is why we provide a new protocol Bank Intra Routing Protocol (BIRP) that intra routes all messages that are exchanged between the main branch and all its branches. The main goal of this routing protocol is to make all branches centralized to the main branch, to save message costs transferred between branches, to decrease delay, and to provide better connectivity.

The rest of the paper is organized as follows. Section 2 provides an overview of SWIFT background and history. Section 3 presents BIRP or Bank Intra Routing Protocol for intra routing system. Section 4 reports the experimental results. And section 5 presents the conclusion.

II. SWIFT BACKGROUND AND HISTORY

SWIFT is a wide financial industry, which provides a secure message exchange for around 8000 financial institutions in more than 200 countries. It always attempts to provide the best services to its customers and this is clear from the products it offers [12], the security it maintains, and the reliability it provides. In Belgium 1973, SWIFT was established by a group of people supported by 239 banks in 15 countries [1]. The Society for Worldwide Inter bank Financial Telecommunication SWIFT started the goal of creating a worldwide message processing. At that time all communications where via telex and there was not very strong security. In 1974, large organizations planned to be members of SWIFT. Initially, it has emphasized its security and reliability. In 1977, SWIFT started to be active and the first message was sent by King Albert. In that year about 3,000,000 messages were exchanged between 518 banks in 21 countries. In 1978, the first ten million messages were sent in less than 12 months.

R. A. Haraty and R. Sadek are with the Division of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon (e-mail: rharaty@lau.edu.lb)

I. Lee is with the Department of Science and Mathematics University of Texas of the Permian Basin Odessa, TX 79762, USA (e-mail: Lee_i@utpb.edu)

1-4244-0941-1/07/$25.00©2007 IEEE

In the 1980s, Hong Kong and Singapore were ready to exchange messages live. SWIFT also introduced the ST100 interface application [1] [2], which was used by 900 customers. With the SWIFT geographical expansion, it introduced new message standards. In order to support traffic growth, SWIFT provided satellite communications for enhancing the financial services.

In 1991, SWIFT received the Computerworld Smithsonian Information technology Award for its excellent work in maintaining financial telecommunication in an advanced way. In 1995, SWIFT opened its office in Germany. Its target was to improve security and reduce costs for customers. In 1996, about 3,000,000 million messages were processed every day.

In 2001, SWIFT's aim was to gain the trust of all banks worldwide and to process messages throughout the net; thus, SWIFT net was established. In 2002, SWIFT net Release 4.0 went live and concurrently the first SWIFT net message was sent [8].

Nowadays most of the banks around the world communicate with each other throughout SWIFT in order to transfer money worldwide in the most secure way. Messages are authenticated through bilateral keys [4], [7] and in this way both parties can legally transfer money to each other.

In the annual report published on 11 August 2006, Johan Kestens [9], the head of marketing and member of the SWIFT Executive Steering Group, said, "We are encouraged by the results customers continue to value our performance in the critical areas of security, reliability and standards, with scores of 9.0 or higher. However, we should not be complacent; the results also pointed to a number of areas for improvement."

On Monday 26 September 2005, the launch of the first international money exchange company located between Western Europe and East Asia took place, which is known by Dubai International Financial Exchange (DIFX) [10]. Their aim is to become the leading message exchange. Their aim is to have customers from all banks over the world. According to Simon Eacott [11], Managing Director of Corporate Banking Services "SWIFT is becoming the 'Microsoft' of financial messaging – the benchmark against which everything else is measured." According to Lori Hricik [11], Executive Vice President and Head of Treasury Services "SWIFT is a critical part of the value chain we deliver to our clients. When thinking about the role that SWIFT plays for them, my paramount concern is the need for resiliency and efficiency."

A. SWIFT Alliance Access

SWIFT Alliance Access (SAA) is a financial message interface, built in a way in order to facilitate the LAN and WAN network communication and transfer money securely. SAA is made up of several applications; each has its importance and its specific role. For example, the routing application has to deal with routing rules and the processing of messages, event journal application's role is to check any action that has been done by the operator in the system, message approval's role is to verify and authorize messages, system management's role is to create operators and start/stop components, etc.

Banks all over the world are directly connected to swift through SAA interface via different wired and wireless lines including dial up, leased line, ISDN, Internet, microwave, VSAT, etc.

SAA runs on the server using the Windows 2000 or XP platform where operators work at different machines and communicate with the central server. The central server is the one that controls the message flow. In SWIFT, messages are classified into two sub formats: 1) Input: every message created at the sender side is called input message, which is forwarded to the correspondent; and 2) Output: messages that are sent to the correspondent are called output messages. The structure of the SWIFT messages is as follows:

1-Header
2-Text block
3-Instances
4-Interventions
5-Appendix

Header

It is the essential part of the message, which consists of the sender and correspondent bank identifier code [2] [3]. The Bic [14] - Bank Identifier Code - is used in order to uniquely identifies the bank and differentiate it from other banks. It is made up of 11 characters as in: JTBKLBBEXXX

The first four characters for the bank name, fifth and sixth character for country code, seventh and eighth character for city, the last three for branches for example XXX is the main branch and B01 for a branch.

Each message has its own UMID - Unique Message Identifier - which is automatically generated by the system. It is used internally by Alliance as the identifier of the message. It forms the primary key of the message. It has the following form: IJTBKLBBEXXX103REFERENCE. The first character is for the sub format, next for the correspondent bic code then the message type and finally the reference of the message.

Text Block

The text block contains the text details of the message. Each message type has its own text block syntax. It is made up of many field numbers and each field has its syntax:
20 is for the sender reference,
32 is for the amount transferred,
59 is for the beneficiary customer name and address, and 52 is for the ordering institution.

Instance Types

The swift message is made up of three instances: original, copy and notification message instances. When a message enters the system an original message instance is created. An important role of the Alliance is that messages can have copies and their process is independent from the original message. They can be created by specifying in routing rules. This allows, for example, an original message to enter the system, to be processed and then completed, and at the same time for a copy to be processed to another queue.

Intervention

All events that occur in the life of the message from the time it is created till its completion will be reported as interventions inside the message. These automatically generated by the Swift Alliance.

Appendix

The main purpose of appendix is to record all network interactions such as a message has been failed to be authenticated, a message has been acknowledged, or a message was rejected so it returns a non-acknowledge, etc. Every reception from a network and every emission to a network must have one appendix attached to the message in order to record this interaction.

The Life Flow of a Message

A swift message, during its life, passes through many queues before being routed to SWIFT. It is first created in the message creation queue, then it is moved to message verification, and finally to authorization. If the authorizer found that a message should be modified, then it will be moved to the modification queue; otherwise, it continues to Sitoswift queue and from there it will be finally processed to swift. The life flow of the message is shown in figure 1.

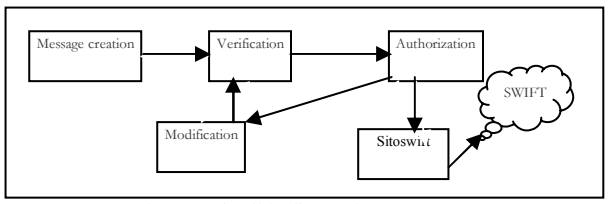

Fig. 1. The life flow of a message.

When a message is forwarded to swift, it is an input message. After an output message has been created swift sends it to the correspondent returning either an acknowledgment or non-acknowledgment to the sender. The original message stays live till a notification acknowledgment or a non-acknowledgment is received by swift.

There are 999 types of swift messages; each one has its own syntax. Some which are financial, acknowledgment, confirmation and free text messages.

SWIFT Alliance Access Services

The SWIFT Alliance interface application provides several services to users and to the interface applications itself:
- Access control application: is the alliance interface,
- Bank information: used in order to search for bank details such as bank name,
- Message processing: for checking and routing messages,
- Routing definition: deals with queues and routing rules,
- Security definition: deals with creating operators and changing passwords, and
- System management: for start and stop components.

Access Control Application

The Access Control application is the key into all SWIFT Alliance applications. It controls all SWIFT Alliance applications and functions. When the user log on into SWIFT Alliance, the applications that the user can use appear in the Access Control window.

These are all the applications that are provided by SAA. The user profile defines which applications to use. The user cannot sign on to SWIFT Alliance unless the administrator has allowed him/her to use the Access Control application. The Security Officer or System Administrator is responsible for creating user operators and gives them privileges through the Security application in order to define the user's profile. Users are not permitted to access the interface application unless they have authenticated usernames and passwords.

Calendar Application

The Calendar application is used to create a yearly calendar, modify a yearly calendar, and schedule Swift alliance applications in order to run them on different times. For example, all messages sent and received are stored in alliance and in order to avoid the database from growing too large, it is better to archive every time by scheduling the message file application. This can be done either manually or automatically.

Correspondent Information File Application

The Correspondent Information File (CIF) [12] [14] contains essential data about correspondents. Each correspondent includes details such as the correspondent bic code, bank name, city, country, address, and so on. The Correspondent file application is used to search for banks based on specific criteria such as bank name, add records, remove records, and modify records that are in the list. The CIF is updated every four months to include banks added to the Alliance.

Event Journal Application

All actions done by users, and by the SWIFT Alliance system, are recorded as events in the Event Journal application. This file provides all actions performed in SWIFT Alliance. Every event has number of details such as the date and time of the event, the name of the user that is using the system, and the importance of the event. The user can search for events in the event journal application based on specific criteria. Events can be implemented as alarms. Alarm events are used in order to alert the operator that a serious action must take place.

Message File Application

The message file application is used to display all incoming and outgoing messages that are in the alliance database. Every message is composed of many instances: original, copy, and notification. Each instance has many interventions, which are the history of the instance and many appendixes that are the network interactions of the message. In this application the user can look to the message details including interventions, and appendixes, complete a message, route a message, reactivate a message, move a message, and archive all messages that are completed in order to save database space.

Monitoring Application

This application is used in order to control the SWIFT Alliance system and ensure that it is running smoothly. If any problem or error occurs, the user can control and manipulate it. The Monitoring application provides an updated status for message queues, events, and processes. For example, it allows a user to check how many messages are in a specific queue in order to avoid being full, or the administrator to check which user is accessing which queue.

Routing Application

The flow of messages within the SWIFT Alliance is controlled by a routing schema. The Routing application is composed of many queues where messages accumulate. Each queue can have many routing rules defined by the user; they are used in order to control the flow of messages. The routing schema is activated when no more changes are needed in the routing rules. In the routing application the user can create routing rules, delete routing rules, activate routing schemas, and deactivate routing schemas.

Security Definition Application

The role of Security Definition application is to define which SWIFT Alliance application a user can access in order to work on it. This is done by assigning a user profile to each user. The Security Definition application is used to create SWIFT Alliance users, modify SWIFT Alliance users, remove SWIFT Alliance users, approve user in order to be able to sign on to alliance, and configure security parameters, such as the number of days which a user password has to change the password. The user that is created must be approved by the administrator in order to have access to the alliance application.

System Management Application

The role of System Management application is to control SWIFT Alliance. It is used to change date/time formats, shut down the system, create queues, back up and restore SWIFT Alliance database, stop and start a protocol application, and define events that need to be set as alarms.

BK Management Application

The main job of bilateral management key [19] [28] is to authenticate messages that are created by the sender and that is sent to the receiver via swift net. Sets of bilateral keys are used by correspondents to legally exchange messages. When the user sends a message, Swift alliance uses the bilateral key agreed on in order to calculate the MAC and attach it to the message. After the receiver receives it, the MAC is calculated based on the bilateral key. If they match, then this means that the message arrived without errors, while if

the MAC did not match, this means that the message is not reliable and it was invaded.

III. BANK INTRA ROUTING PROTOCOL

Intra Routing Protocol was created using the alliance developer kit [3] for intra routing system to take every input message from the routing point (BIRP Input) and checks if the Bic Receiver [5][6] is a branch or not. Now there are two possibilities: 1) If the correspondent is a branch then it creates two messages: an output message, which is sent to queue Output. It is processed to the correspondent and an acknowledge notification instance sent to queue Acked informing the sender that the message has been received successfully by the correspondent; 2) If the receiver is not a branch then only an instance non-acknowledge is created and the message is routed to queue Nacked so that it can be forwarded to Sitoswift by a routing rule specified on this queue. The message is then directed to SWIFT for processing an output message to the correspondent. The original instance will always be complete. It is free from any routing point while both the notification and the output messages are live, and placed in the corresponding queues BIRPAcked and BIRPOutput. All events are reported as interventions inside every message.

There are four different BIRP routing points created in order to handle the incoming messages in an efficient way. They are listed as follows: BIRPinput, BIRPoutput, BIRPacked, and BIRPnacked. The BIRPinput is the central queue of the component where every incoming message is read by the BIRP protocol. Routing rules have been created on this queue so that if the message instance is an acknowledgment, then it is sent to queue BIRPacked, while if the message instance is a non-acknowledgment then it will be sent to BIRPnacked and the output message will be sent to BIRPoutput in order to be forwarded to the correspondent by a specified routing rule.

All messages that pass through these routing points are automatically handled by the message processing function whose role is to process and control messages. The BIRP protocol is integrated in the main branch system and all branches are linked to it. All messages will pass through the main branch and its role is to decide whether to create an output message and transfer it to the branch or to forward it to SWIFT so that it can send the message to the specified correspondent. This protocol can be started or stopped by the system administrator. In this way all messages that are exchanged between branches will not be forwarded to SWIFT. All the communication will take place between the branches and the main branch. In the case where the receiver is a branch, the routing will be intra within the main branch. The message will go from branch1 "sender" to the main branch, it is verified and authorized and then it enters the BIRPinput queue where the protocol starts to scan the message. Since the correspondent is a branch "branch2", an output message is created and processed to queue BIRPoutput, which will then be forwarded to branch2. An acknowledge notification will be created confirming that the message has been received successfully by branch2.

Now in the second scenario, the receiver is not a branch. The BIRP routing protocol creates a non-acknowledgement notification sending it to the BIRP Nacked queue. By a defined routing rule, the message will be forwarded to SitoSwift queue and then it continues its tour to SWIFT.

IV. EXPERIMENTAL RESULTS

We applied the BIRP routing protocol to ECOBANK of AFRICA, which is the largest bank in Africa. It has branches in 12 countries. The following results were achieved:

A. Cost ($)

A message sent charges 0.25 cents per day. If the bank sends 9000 messages per day all over the world and around 4000 of them are for their branches in different countries, then we calculate the annual costs in order to see how much savings per year. Annual Message Costs for Every Branch =
4000 messages *0.25 = 1000 EUR per day * 317 * $1.28 = $405.760 saving costs per year.

As we can see the increase of branches leads to the increase of saving costs to the bank. In ECOBANK there are 12 countries worldwide so in this case 12 * $405.760, which is around 5 million dollars a year is clearly worth to be saved.

Comparing Costs Before and After using BIRP

If we take a look at the expenses without the intra protocol we find that the total yearly expenses for every branch connected to SWIFT is as follows:
Maintenance + SAA License + Security Infrastructure = 40,000EUR = $51,200
Total Yearly Expenses = $51,200 + $405,760 messages = $456,960 for each branch

While the expenses for the new BIRP routing protocol are:
Total Yearly Expenses = 5000 EUR License + 0 (since no more message costs with the new intra routing system)

We can see the difference in message costs saved and can conclude that it is worth using BIRP because no security infrastructure is needed since it is already

implemented in the main branch. If we carry out simple projections over the next five years, we can see in figure 2 that 1,260,000 messages are sent yearly to branches and these cost 405,760$ per year. In 2010, we are expecting the bank to save over 2 million dollars, which is significant. This corresponds to 6,340,000 messages and this is all for one branch.

According to ECOBANK around 25 million dollars are saved in 5 years for 12 countries. This is excellent savings for the bank.

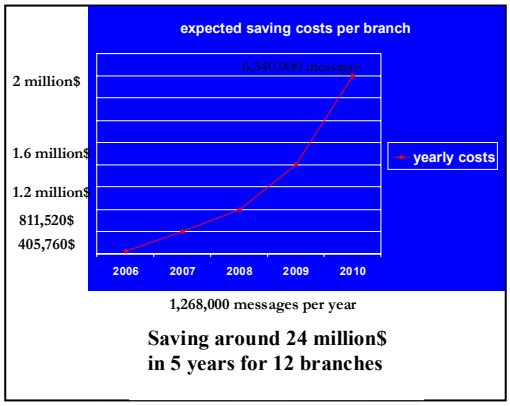

Fig. 2. Expected savings per branch.

B. Delay

The estimated time taken for a message to go to SWIFT then forwarded to the correspondent takes an estimated minimum of 3 to 5 seconds depending to the network traffic, while with BIRP less than 1 second because of real time exchange between branches. There are several aspects that affect delay: distance and physical medium.

In SWIFT, the sender is connected to SWIFT through routers, servers, VPN boxes and the delay is increased with the increase in distance, while there is no such delay in BIRP protocol since the connection is direct.

C. Connectivity

BIRP is a good solution for connectivity since branches will not be directly connected to Swift. If a SWIFT failure occurs, BIRP will still operate normally. While if branches are connected to SWIFT, then any failure in the connectivity will stop the message processing until the connection is restored.

As can be seen in figure 3, in case 1, which is the BIRP protocol all messages are connected to the main branch, while in case 2, all banks and branches are connected to SWIFT.

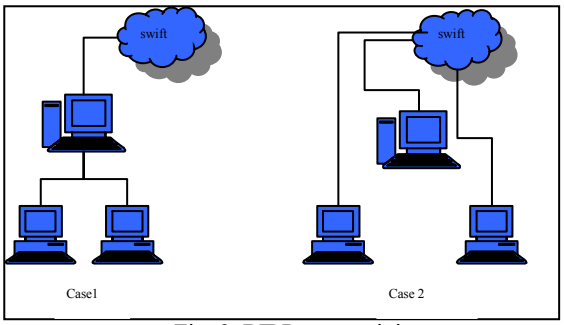

Fig. 3. BIRP connectivity.

V. CONCLUSION

This paper presented BIRP, which is a major enhancement over the traditional SWIFT protocol. BIRP proved its power in message exchange. Specifically, BIRP showed that there is no need for additional security infrastructure. BIRP also showed that messages can be received by branches in less delay because of a shorter distance since there is a difference if a message flows through SWIFT or only through the main branch. BIRP is also good on connectivity.

The paper also presented an analysis of BIRP protocol and how it efficiently works. Comparing to yearly costs, it was demonstrated that it is worthy to use the BIRP routing protocol since banks can find this protocol as an asset for their investment. Finally, we can say that BIRP can be considered an efficient and dynamic real time exchange protocol for intra routing banks.

REFERENCES

[1] SWIFT History, SWIFT routing protocol [Online]. Available: http://www.SWIFT.com. 2007
[2] SWIFT Team Belgium (2004).Getting Started with SWIFT Alliance Access (Release5.5) version 2.0.
[3] SWIFT Team (2005). Alliance Developer Kit ADK.
[4] SWIFT Team. BESS Bilateral Key Exchange. 2006.
[5] Bic Directory [Online] Available http://www.SWIFT.com/biconline/. 2007.
[6] SWIFT Team (2005). Bic Directory.
[7] BKE Bilateral Key Management [Online] Available http://public.logicacmg.com/~fsd/Prod_descr/Bke.PDF. 2007.
[8] SWIFT Team. SWIFT Net Phase 2. 2007.
[9] Making it easier to do business with SWIFT, [Online]. Available: http://www.SWIFT.com, 11 August 2006.
[10] Dubai Bridges the Gap between East and West, [Online]. Available: http://www.SWIFT.com, 15 November 2005.
[11] Kumar Parekh. Society for Worldwide Inter bank Financial Telecommunication. 2006.
[12] SWIFTAllianceproducts [Online] Available http://www.perago.com/Products/services/SWIFT.html.2007.