DESIGN CONSIDERATIONS FOR FINANCIAL INSTITUTION INTELLIGENT NETWORKS

by

Naum Goldburt

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

2004

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Copyright

Approval

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

(signature) 4/27/04 (date) Professor Syed V. Ahamed
Chair of Examining Committee
City University of New York

(signature) 04/27/04 (date) Professor Theodore Brown
Executive Officer
PhD Program in Computer Science
City University of New York

Professor Michael Anshel
Committee Member
City University of New York

Professor Michael Kress
Committee Member
City University of New York

Distinguished Member of Technical Staff
Dr. Monima Briggs
Committee Member
Lucent Technology

Supervisory Committee

The City University of New York

Abstract

DESIGN CONSIDERATIONS FOR

FINANCIAL INSTITUTION INTELLIGENT

NETWORKS

By

Naum Goldburt

Advisor: **Professor Syed V. Ahamed**

Financial Institution Intelligent Network (FIIN) were constructed as Intelligent Peripheral (IP) with entire Intelligent Network (IN) architecture specifically designed for routing and transmission of financial messages in context of Advanced Intelligent Network (AIN).

Intelligent Peripheral adaptive infrastructure with accumulated intelligence can accommodate the specific needs of financial industry with corporate electronic banking. FIIN offers a single window to the world of networks and makes the network selection process transparent to its applications. FIIN provides network services to connect to the variety of the highly specialized financial networks such as SWIFT, FedWire, CHIPS, CHAPS, TARGET, BoJNet. Federal (or state-owned) banks and Clearing Houses offer country-mandatory and currency-specific proprietary networks whereas SWIFT

provides the international network that become de-facto standards for respective financial institutions worldwide.

All AIN basic building blocks are employed in conjunction with FIIN messaging systems: Message Switching Point (MSP), Message Transfer Point (MTP), Message Control Point (MCP). Described Message Control System (MCS) that monitors message flows within FIIN and between FIIN and AIN components. Shown protocols between MCS and SS7, links between MTP and Service Transfer Point (STP) of AIN. Demonstrated MCS separation of data from control information, namely financial messages are processed separately from system messages, secret keys, certificates, confirmations, acknowledgments.

Location (x) and application (y) independence allows easy adoption of new services and products (Service Creation Environment) anywhere anytime. Four application sub-layers simplify the message flow architecture providing for highly flexible prioritization mechanism using queuing models.

FIIN communicates with retail and wholesale business customers in any combination between Business (B) and Financial institution (F): B2F, F2F, B2B.

FIIN is successfully implemented for a major bank. Described functionality of a real FIIN system.

Preface

Ten years of full time work in the area of Message Services Applications and successful implementation of brand new software creative products led me to believe that I can summarize my results to be used by many companies. The years of parallel study in CUNY and the latest classes of Professor Ahamed added greatly to my confidence that the goal can be achieved. I was deeply influenced by Professor Ahamed's results in Advanced Intelligent Networks and encouraged by him to proceed with my own research.

I want to express my great appreciation and acknowledgments to Professor Syed Ahamed for his continuous support and guidance.

TABLE OF CONTENTS

COPYRIGHT ..... **II**

APPROVAL ..... **III**

ABSTRACT ..... **IV**

PREFACE ..... **VI**

1 INTRODUCTION **1**

1.1 CORPORATE ELECTRONIC BANKING 2
1.2 OVERVIEW OF CONTEMPORARY FINANCIAL NETWORKS 10
1.2.1 Major Financial Networks *10*
1.2.2 Growth of the SWIFT Network *13*
1.3 FINANCIAL INSTITUTION NETWORKS ARCHITECTURE 15
1.3.1 Advanced Intelligent Networks (AINs) *18*
1.3.2 Financial Institution Networks *22*
1.4 TRENDS IN DESIGN OF FINANCIAL INSTITUTION NETWORKS 23
1.4.1 Accumulated Intelligence *30*
1.4.2 Integration *35*
1.4.3 Shared Use *37*

2 ARCHITECTURE OF FINANCIAL INSTITUTION INTELLIGENT NETWORK (FIIN) **38**

2.1 HYPOTHETICAL MODEL FUNCTIONAL REQUIREMENTS 38
2.1.1 Basic Structure of Queuing Model *40*
2.1.2 Queuing Model Selection *51*
2.1.3 Given Financial Messaging Requirements *58*
2.1.4 Performance and Capacity Planning *59*
2.1.5 Existing Messaging Systems *65*
2.1.6 SWIFT, TELEX, and FAX *73*
2.1.7 Trends and Business Drivers *75*
2.1.8 Special Cases *99*
2.1.9 Manual Message Processing *108*
2.1.10 Monitoring and Control *118*
2.1.11 Daily Checkpoints *129*
2.1.12 Summary of the Requirements *132*
2.1.13 Flowchart Examples *134*
2.2 BUILDING BLOCKS OF FINANCIAL INSTITUTION INTELLIGENT NETWORK 144

2.2.1 Financial Institution Intelligent Network as Intelligent Peripheral *144*
2.2.2 Bank Messaging System *148*
2.2.3 Application Sub-layers *151*
2.2.4 Basic Building Blocks of Bank Messaging System *154*
2.3 MESSAGE SWITCHING POINT (MSP) 157
2.3.1 Four Layers of Switching *157*
2.3.2 Application Interface Switch *179*
2.3.3 Application Ready Switch *199*
2.3.4 Message Ready Switch *200*
2.3.5 Network Selection Switch *202*
2.4 MESSAGE TRANSFER POINT (MTP) 204
2.4.1 Functions at Each Switching Layer *206*
2.4.2 Links Between Message Transfer Point and Carrier's Service Transfer Point at Network Selection Switch *207*
2.5 MESSAGE CONTROL POINT (MCP) 217
2.5.1 Hardware and Software Components *217*
2.5.2 Server Data Bases *220*
2.6 OPERATION, ADMINISTRATION AND MAINTENANCE 221
2.6.1 Operating System *221*
2.6.2 Service Management System *223*
2.6.3 Data Base Administration *224*
2.6.4 Service Creation Environment *225*
2.6.5 Service Administration *225*
2.6.6 User Access Administration *226*
2.6.7 System Access Administration *227*
2.7 DATA SECURITY AND INTEGRITY 227
2.7.1 Line and Data Encryption *230*
2.7.2 Network Login Key Infrastructure *230*
2.7.3 Message Authentication *230*
2.8 MESSAGE CONTROL SYSTEM (MCS) 231
2.8.1 Message Control Link *232*
2.8.2 Message Link *240*
2.8.3 Administrative Data Link *240*
2.8.4 Separation of Control and Data Messages *240*
2.8.5 Control Channel *241*
2.8.6 Line Utilization *241*

3 INTERFACE BETWEEN FINANCIAL INSTITUTION INTELLIGENT NETWORK AND CARRIERS 244

3.1 PRIVATE AND PUBLIC CARRIERS 244
3.2 TRANSMISSION LINES 253

4 PROTOCOLS IN FINANCIAL INSTITUTION INTELLIGENT NETWORKS 256

4.1 INDUSTRY STANDARDS 256
4.2 PROPRIETARY PROTOCOLS WITH PRIVATE CARRIERS 261
4.3 PROPRIETARY PROTOCOLS WITH PUBLIC CARRIERS 264

5 FUNCTIONALITY OF FINANCIAL INSTITUTION INTELLIGENT NETWORK 267

5.1 SENDING AND RECEIVING FINANCIAL MESSAGES 267
5.1.1 Data Entry and Graphic User Interface 267
5.1.2 Internal Routing 271
5.1.3 Application Interfaces 271
5.1.4 Charging, Statistics and Archiving 272
5.2 SERVICE APPLICATIONS 272
5.2.1 Software Defined Internal Network 272
5.2.2 Message Scanning 272
5.2.3 OFAC Scanning 273
5.2.4 Message Reconciliation and Matching 276
5.2.5 Message Re-advising 276
5.2.6 Continuous Linked Settlement (CLS) 276

6 CONCLUSION 279

6.1 CONTRIBUTION 279
6.2 PROSPECTIVE 281

7 APPENDIX 283

7.1 APPENDIX A 283
7.1.1 Data Entry Function 283
7.1.2 Special DEV Features 287
7.2 APPENDIX B 292
7.2.1 Retrieving a Draft Message into Data Entry Queue 292

BIBLIOGRAPHY 296

GLOSSARY 299

TRADEMARKS 303

INDEX 304

List of Tables

Table 1 Performance Formulas for Exponential Queuing Model 48
Table 2 Performance Measurements for M/M/s Queuing Model 49
Table 3 Performance Measurements for Non-exponential Queuing Models 50
Table 4 BMS Interfacing Applications 150
Table 5 BMS Applications. 151
Table 6 Queue Name Interpretation. 158
Table 7 Bank's Branch Country and City Codes. 159
Table 8 Queue Functions. 160
Table 9 Queue Direction. 161
Table 10 BMS Module Names. 161
Table 11 BMS Module Types 161
Table 12 MQ Names Format. 197
Table 13 MQ GPS Names Example. 198
Table 14 CBT – SWIFT FIN Interface 216
Table 15 Comparison of Typical Voice Call, ISDN Data Call, and FIN Connection. 239
Table 16 BMS Communications Primitives. 253
Table 17 Charges Related to Connection to POP. 255

List of Illustrations

Figure 1 Concept of B and F Bilateral Relationships. 5
Figure 2 Payment Instructions. 6
Figure 3 B2B Integration of Internet in Financial Network Services. 9
Figure 4 SWIFT FIN 10-year Traffic Evolution. 14
Figure 5 Rudimentary SNA Computer Network. 16
Figure 6 Advanced Intelligent Network Platform 0. 19
Figure 7 Software Defined Network 20
Figure 8 SWIFT Architecture. 25
Figure 9 Logical SWIFT Message Flow. 29
Figure 10 B2B SWIFT Next Generation Approach. 30
Figure 11 Single Window Concept 36
Figure 12 Basic Queuing Process 41
Figure 13 Queue Performance Measurement for GI/G/1 Model 54
Figure 14 BMS Queuing Network Load Sample 57
Figure 15 Input to FedWire/CHIPS (Fragment 1) 135

Figure 16 Input to FedWire/CHIPS (Fragment 2)..... 136
Figure 17 Input to FedWire/CHIPS (Fragment 3)..... 137
Figure 18 Acknowledgement on Input to FedWire/CHIPS (Fragment 1). 138
Figure 19 Acknowledgement on Input to FedWire/CHIPS (Fragment 2). 139
Figure 20 Acknowledgement on Input to FedWire/CHIPS (Fragment 3). 140
Figure 21 Output from FedWire/CHIPS (Fragment 1). 141
Figure 22 Output from FedWire/CHIPS (Fragment 2). 142
Figure 23 Output from FedWire/CHIPS (Fragment 3). 143
Figure 24 Multiplatform Computer Network. 145
Figure 25 Financial Institution Intelligent Network. 147
Figure 26 Bank Messaging System. 149
Figure 27 Application Sub-Layers..... 153
Figure 28 Message Control Point. 155
Figure 29 Four Layers of Switching. 162
Figure 30 BMS Multiple Instances Scenario..... 163
Figure 31 Input to Network (Fragment 1). 164
Figure 32 Input to Network (Fragment 2). 165
Figure 33 Input to Network (Fragment 3). 166
Figure 34 Input to Network (Fragment 4). 167
Figure 35 Input to Network (Fragment 5). 168
Figure 36 Acknowledgments on Input to Network (Fragment 1). 169
Figure 37 Acknowledgments on Input to Network (Fragment 2). 170
Figure 38 Acknowledgments on Input to Network (Fragment 3). 171
Figure 39 Acknowledgments on Input to Network (Fragment 4). 172
Figure 40 Output from Network (Fragment 1). 173
Figure 41 Output from Network (Fragment 2). 174
Figure 42 Output from Network (Fragment 3). 175
Figure 43 Output from Network (Fragment 4). 176
Figure 44 Output from Network (Fragment 5). 177
Figure 45 Output from Network (Fragment 6). 178
Figure 46 Application MQ Interface Receive Process - Input to Network. 182
Figure 47 Application MQ Interface Acknowledgement (Rejection) Process - Input to Network. 183
Figure 48 Application MQ Interface Send Process - Output from Network..... 184

Figure 49 Application MQ Interface Reply Process – Network Input/Output (Fragment 1)..... 185
Figure 50 Application MQ Interface Reply Process – Network Input/Output (Fragment 2)..... 186
Figure 51 Application MQ Interface Reply Process – Network Input/Output (Fragment 3)..... 187
Figure 52 Network MQ Interface Send Process – Input to Network (Fragment 1). 188
Figure 53 Network MQ Interface Send Process – Input to Network (Fragment 2). 189
Figure 54 Network MQ Interface Receive Process – Output from Network (Fragment 1). 190
Figure 55 Network MQ Interface Receive Process – Output from Network (Fragment 2). 191
Figure 56 Network MQ Interface Receive Process – Output from Network (Fragment 3). 192
Figure 57 Network MQ Interface Rejection Process – Output from Network (Fragment 4). 193
Figure 58 Network MQ Interface Reply Process – Network Input/Output (Fragment 1). 194
Figure 59 Network MQ Interface Reply Process – Network Input/Output (Fragment 2). 195
Figure 60 Network MQ Interface Reply Process – Network Input/Output (Fragment 3). 196
Figure 61 Interconnection of STPs in the CCS7 Signaling Network... 205
Figure 62 Example of SMS Function Selection Screen. 224
Figure 63 Secured Network Access..... 229
Figure 64 MSP Integration into CCS7 Signaling Network. 235
Figure 65 Acknowledgment Rate. 243
Figure 66 Dual Line Connection to NAPs..... 245
Figure 67 PVC Data Flow. 246
Figure 68 Single Window Detail. 247
Figure 69 MSP Normal Operations. 248
Figure 70 MSP Switch in Case of VPN/Router/Line Failures 249
Figure 71 MSP Switch in Case of Network Gateway Failue 250
Figure 72 Connection to POP. 254
Figure 73 OSI and SS7 Protocols. 257
Figure 74 Comparison of Communication Protocols. 258
Figure 75 Virtual Circuit Number Assignment. 260

Figure 76 Application Header Format for Input to Network in the Protocol Between FIIN and SWIFT. 262
Figure 77 Application Header Format For Output from Network in the Protocol Between FIIN and SWIFT. 263
Figure 78 Application Header Format for Input to Network in the Protocol Between FIIN and IC. 265
Figure 79 Application Header Format for Output from Network in the Protocol Between FIIN and IC. 266
Figure 80 DEV Message Flow. 269
Figure 81 OFAC – Input to Network 274
Figure 82 OFAC – Output from Network. 275
Figure 83 Continuous Linked Settlement. 277
Figure 84 The CLS Bank. 278

1 Introduction

Corporate world began to enter the world of telecommunications in the early 1980s when TCP/IP was just standardized with Internet infrastructure. In the 1990s corporate use of telecommunications was developing at enormous acceleration along with the use of the Internet. In the 21st century the use of electronic business reaches the speed of industrial revolution.

Networks accumulate intelligence and are being developed toward Advanced Intelligent Networks (AIN). Corporate networks can be seen as developing toward Intelligent Peripherals (IP) within AIN.

Financial Institution such as banks, brokerage firms, stock exchanges, clearing houses have a special role among the corporations. Banks meet regulations specific to each country and normally require license to operate.

Changes in the business environment and evolution in technology require new solutions in financial message (transaction) processing. For example, SWIFT departs from its traditional proprietary X.25-based Transport Network to TCP/IP-based Internet technologies utilizing newly created XML-based industry standards. New SWIFTNet services include InterAct, FileAct, and more traditional FIN. New SWIFTNet service applications offer value-added network services for financial infrastructures such as Continuous Linked Settlement (CLS), Real-Time Gross Settlement (RTGS). It creates new opportunities as well as challenges to Financial Institutions.

The changing financial world necessitates the research and proposal of new solutions to accommodate the challenges of today's technology.

This research offers the proven solution that can be incorporated in the future developments.

1.1 Corporate Electronic Banking

The Internet is the cost-effective way to carry non-critical information. Despite of the all advantages of the Internet it is not perceived reliable in a sense of time and security to carry mission-critical information where specialized networks will continue to be used and developed. Financial networks provide highly secured, speedy, and very reliable network services to financial institutions. Financial network services and service applications address main problems experienced with the Internet: reliability, congestion, throughput, security.

Any Business (B) involves money transfers in one form or another. In most countries handling money transactions is prerogative of the banks. Banks are Financial Institutions (F, FI) with special license to operate money transfers[1].

[1] Terms "Bank" and "Financial Institution" are used interchangeably in the text.

Business partners are clients of financial institutions. Thus different combinations of bilateral relationships are created. Most common configurations are the following[2]:

- Business to Business (B2B),
- Business to Financial Institution (B2F),
- Financial Institution to Financial Institution (F2F).

The concept of B and F bilateral relationships is shown on Figure 1.

This research concentrates primarily on FIs network services, design and architecture of Financial Institution Networks (FIN), adaptability of their architecture, building blocks, operating and control systems, administrative services and other intelligent components.

FIs are going through substantial structural changes due to banking deregulation, globalization of marketplace, new demands on delivery of any information, anytime, anywhere.

FIs face several technology challenges:

- Straight Through Processing (STP),
- Continuous Linked Settlement (CLS),
- Real Time Gross Settlement (RTGS),
- Transaction Flow Monitoring (TFM),
- Risk Management.

---

[2] Person or Individual communications with B and F (P2B, P2F) are outside of the scope of this research.

Modifications of FIs legacy systems are very costly.

New developments in IN such as adaptive infrastructures provide high volume low cost transaction processing along with financial services.

There are phased evolution of electronic banking in FIs:

- e-commerce web sites for retail customers (retail banking),
- e-business for wholesale customers,
- e-market financial services for wholesale banking.

FIs financial services, also called bank's products, or back-end applications, depend on business clients. Large corporations with annual revenue over $250 million are sometimes called fat clients, small corporations (or businesses) with annual revenue less than $10 million are called thin clients. Self-employed individuals and simply retail customers can also be included in thin client category.

Wholesale banking applications serve fat clients, retail banking applications serve thin clients, hybrid model is used for midrange clientele.

Home banking using bank's powerful web sites represents e-commerce for thin clients, or retail banking.

Wholesale e-business are fully integrated wholesale trades between corporate buyers and sellers. Traders are corporations (wholesale corporate customers) acting via their banks in worldwide internet environments (see Figure 2). Fat clients normally directly connected to their banks via leased

lines (DLC – Directly Linked Customers). E-business requires integrating Internet technologies and re-engineering.

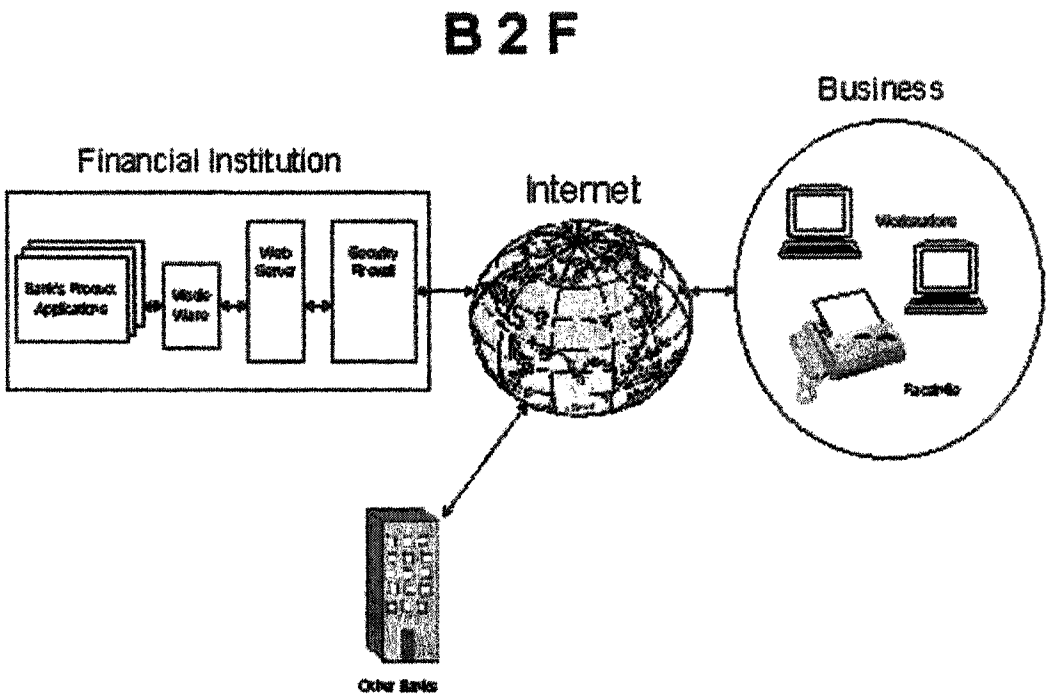


Figure 1 Concept of B and F Bilateral Relationships.

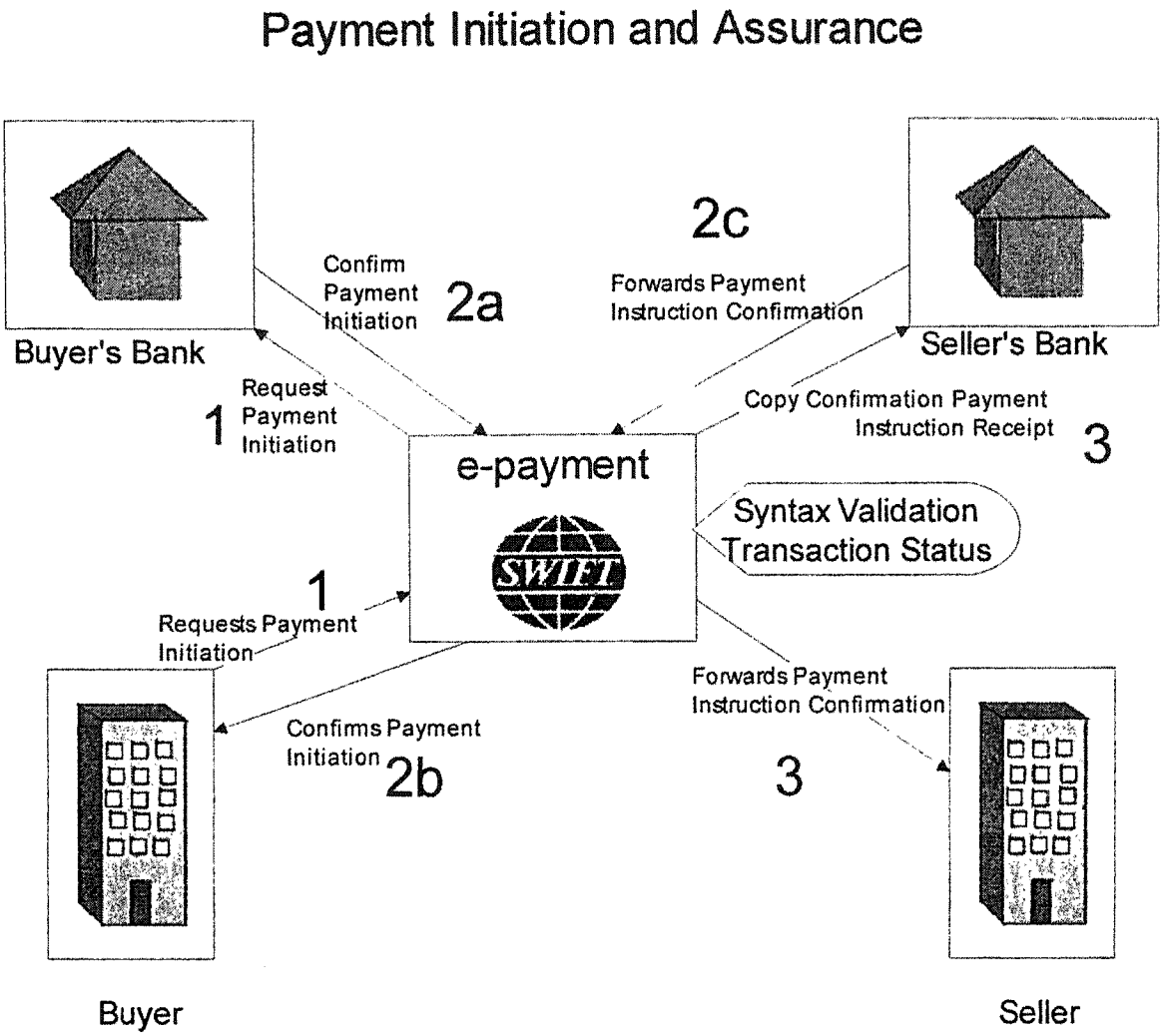


Figure 2 Payment Instructions.

Financial services for e-markets present new challenges to FIs. Corporate customers create conglomerates or alliances to conduct certain segments of their business (for ex. CLS, RTGS, TFM). CLS currently has about 20 member-banks connecting to one CLS bank to reduce the payment settlement risk across the different currencies.

Most banks provide the following financial services:

- electronic payments,

- security settlements,
- foreign exchange,
- cash management,
- commercial lending and finance,
- trade services,
- capital markets,
- corporate trust and custody,
- risk management.

Large banks provide network services such as connection to financial networks: SWIFT, FedWire, CHIPS (country dependent), TELEX, FAX (with additional security like authentication and encryption), external and internal proprietary networks.

Business users consider different factors in selecting the bank and its products:

- rank/reputation,
- reliability,
- security,
- performance,
- capacity,
- availability,
- scalability,

- compatibility.

Approach of Business to Business integration of Internet and Financial Network Services is shown on Figure 3.

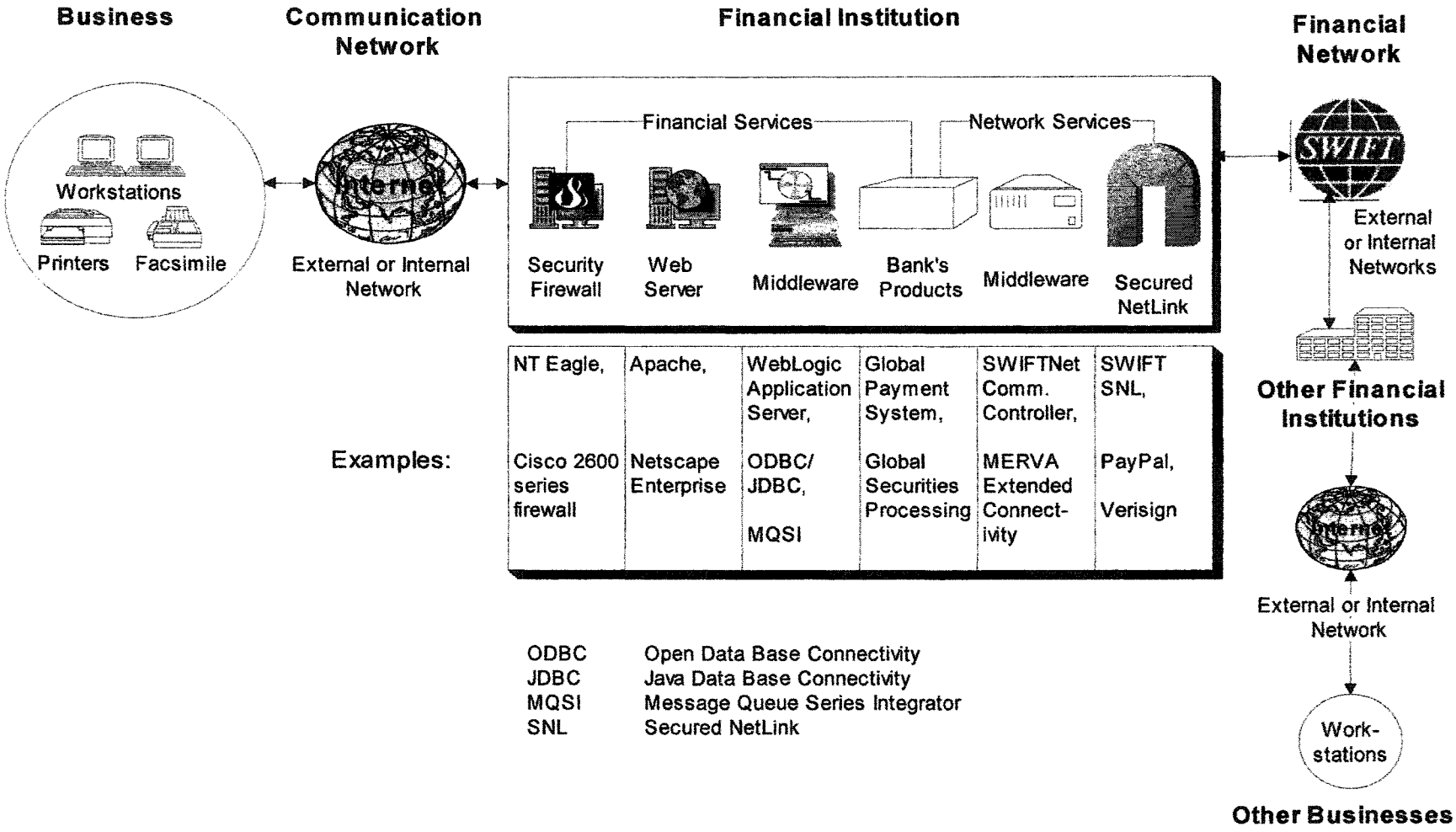


Figure 3 B2B Integration of Internet in Financial Network Services.

1.2 Overview of Contemporary Financial Networks

1.2.1 Major Financial Networks

In the world of financial telecommunications trillions of transactions involving billions of dollars are executed every day. The information required to process these transactions is transmitted as electronic messages via financial networks. Financial networks provide for secured and timely delivery of these transactions as financial messages from senders to recipients. Correspondents are financial institutions like banks, brokerage firms, stock exchanges, clearing houses. The major systems allowing to automate processing of financial transactions are the following [22]:

ACH – Automated Clearing Houses are an electronic alternative to the traditional paper-based check collection system.

EDI - Electronic Data Interchange handles the exchange of routine business agreements such as purchase orders, price quotations and final payments in a series of direct electronic transactions. Value added networks (VANs) are the leading providers of EDI services.

FEDWIRE – is a real-time gross settlement funds transfer network operated by the Federal Reserve Bank. Funds transferred over the FEDWIRE are immediate, irrevocable, and guaranteed. FEDWIRE also serves as the final settlement entity for CHIPS, ACH and check processors. FEDWIRE extended

its operating hours to 18 hours a day to allow the timely settlement of cross currency transactions.

SWIFT – Society for Worldwide Interbank Financial Telecommunications is an international telecommunications network for transmitting and routing financial messages. SWIFT only carries financial messages and does not provide settlement.

CHIPS – Clearing House Interbank Payments System is an on-line, real-time electronic payment system owned and operated by the New York Clearing House Association. The majority of CHIPS transactions involve the US dollar settlement of cross-currency transactions. Settlement is net same day and is conducted through the Federal Reserve Bank of NY.

Credit Cards – method for customers to gain access to the payment system. Card issuers worldwide are banks, retail stores, telephone companies, oil companies, and other financial service providers such as American Express and Dean Witter/Discover.

EFT - Electronic Funds Transfer is the paperless movement of funds between accounts. Most often, ATM are used to provide consumers with cash, and point of sale terminals are used to pay for goods and services at the merchant location.

ATM – Automated Teller Machines are principal delivery system for bank services. Expanded services include ticket purchases, mutual fund

transactions, monthly statements. Regional and national networks have broadened consumer access and increased the convenience of using an ATM.

Debit cards – allow consumers access to funds in their bank accounts at the point of sale. On-line debit transactions are authorized and settled immediately through ATM network switches. They provide immediate verification and transfer of funds from the consumer's account to the merchant's account.

Retail banking delivery – are branch networks and alternate delivery channels such as ATMs, customer activated terminals (CATs), and home banking services.

MERVA – Message Entry and Routing with interfaces to Various Applications is the SWIFT/TLX/FAX service provider for FIs back-end applications.

The most advanced private carrier for financial messages is Society for Worldwide Inertbank Financial Telecommunications (SWIFT). SWIFT does not process the messages i.e. does not do money transfer, securities settlements, etc. SWIFT just transmits and routes financial messages between its correspondents.

Public interexchange carriers (IC) like AT&T, MCI are also used to transmit financial messages via Telex or Facsimile but they are not specifically designed to carry financial messages. Telex is the international 50-bps service used over much of the world.

Financial Networks or Financial Network Service Providers (NSPs) own routers that linked by leased channels from common ICs (such as AT&T). IC provides basic long distance transmission service, NSP adds value to the basic service. Financial NSPs from being simple VANs became PVNs widely used by all financial institutions worldwide.

1.2.2 Growth of the SWIFT Network

SWIFT yearly traffic reached the two billion FIN message mark on 22 December 2003, doubling in volume since the one billion mark was reached in 1999. The daily traffic is expected to reach 10 million messages a day. The average daily value of payment messages was above $10 trillion.

FIN is SWIFT's core store-and-forward messaging service serving over 7,500 financial institutions in 200 countries to exchange financial data securely, cost effectively and reliably.

SWIFT has different categories of members, sub-members, and participants which are banks, broker-dealer corporations, investment management institutions. Working closely with ISO, SWIFT produces de facto standards for financial industry. SWIFT provides network services for the following financial applications that grew substantially in 2003:

- Payment systems, +10%.
- Security infrastructures, +19%.

- Treasury, +18%.
- Trade Finance, +3%.

SWIFT provides the graph of its 10-year FIN traffic evolution:

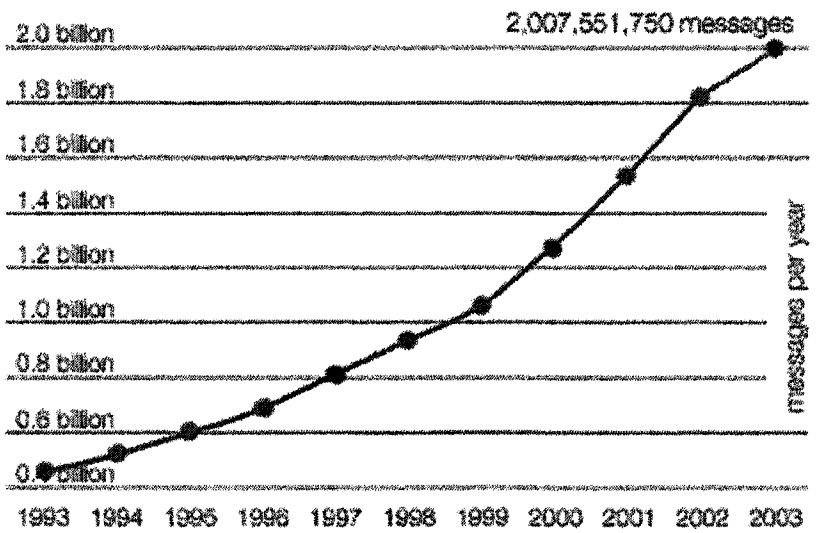


Figure 4 SWIFT FIN 10-year Traffic Evolution.

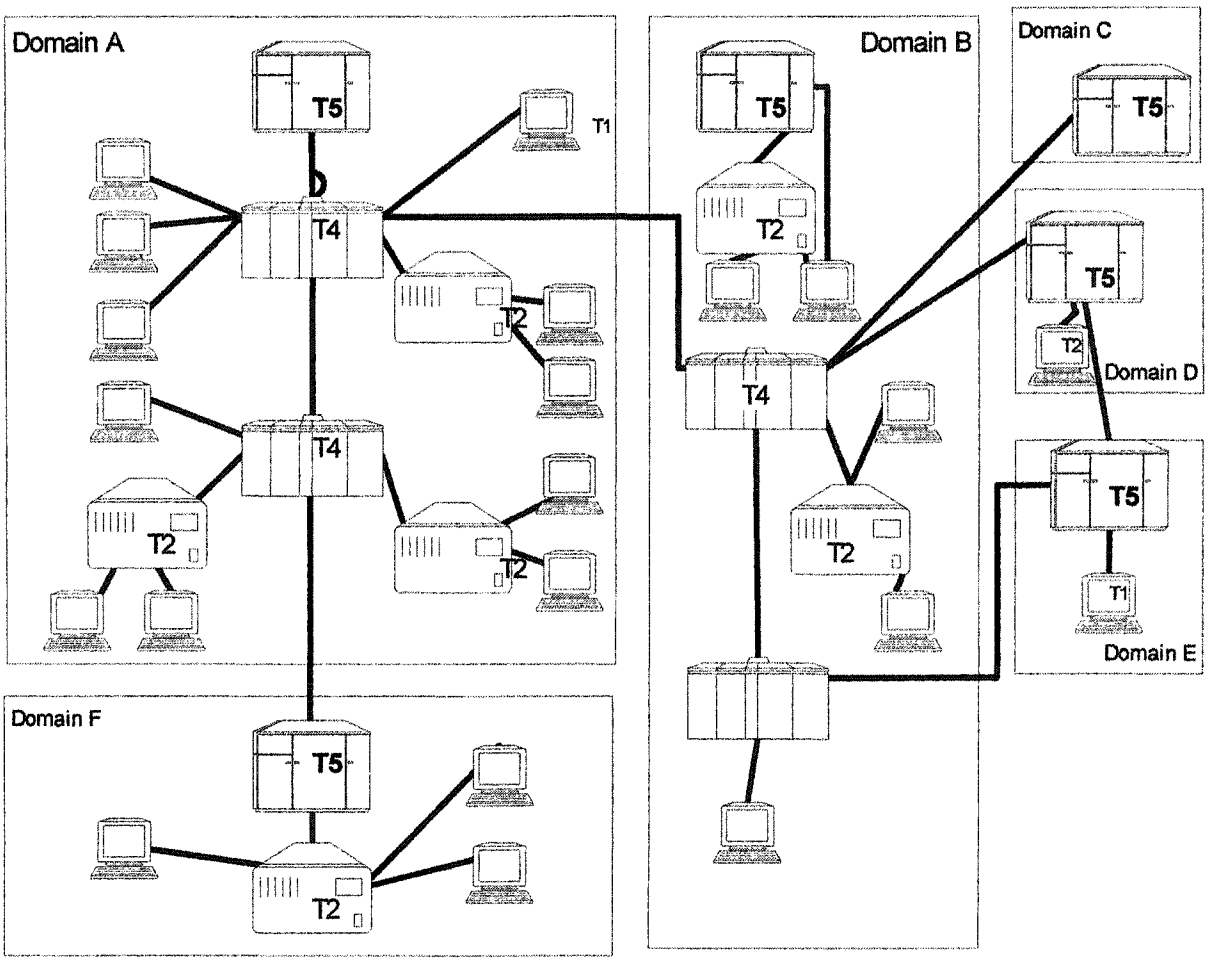
Currently SWIFT is involved in development of new generation Secured IP-based Financial Network (SIPN) embracing Internet technologies. Information about SWIFT is available in [21].

1.3 Financial Institution Networks Architecture

Financial Institution Networks were originally developed as Computer Networks, later merged with Communication Networks, and now heading toward Intelligent Networks.

Computer Networks provide computer-to-computer connectivity with the bursty nature of data transfer. SNA (IBM's System Network Architecture) and DECNET (Digital Equipment Corporation's Network Architecture) are de facto standards and protocols for Computer Networks. DEC in its open VMS (Virtual Management System) adopted and implemented OSI protocols. (DEC was later acquired by COMPAQ which in turn merged with Hewlett Packard.)

Figure 5 shows an example of SNA Computer Network [**8**, **9**]. Each domain is managed by a host node with its own System Service Control Point (SSCP) which combines functions of SSP, STP and SCP of IN. An SNA node contains a physical unit PU that represents the device and its resources to the network. For example, type 2 nodes are user-programmable. PU 2.1 is used in conjunction with LU 6.2 in implementing Advanced Program-to-Program Communications (APPC) in multiplutform environment.



Each domain is managed by a host with its own SSCP.
SNA = System Network Architecture;
SSCP = System Service Control Point;
T5 = host node, mainframe computer that contains SSCP;
T4 = communication controller,
T2 = cluster controller,
T1 = terminal.

Figure 5 Rudimentary SNA Computer Network.

Evolution of Communication Networks from conventional Plain Old Telephone System (POTS) to Intelligent Networks (INs) with common channel signaling facilities (CCS) is described in great detail in [**1**]. The architecture and operation of Public Domain Intelligent Networks (PDIN) has to comply with international CCITT (ITU-T) standards. IN/1 is the first network implemented in the public domain with 800 service. IN/1 provides for the following [**1**, **5**, **6**]:

- alternate billing service (ABS),
- emerging response service (911),
- area wide centrex (AWC),
- pay per view (PPV),
- private virtual network (PVN).

PVNs are of particular interest for Financial Institution Networks (FIN) which are Value Added Networks (VANs) that mostly implemented as PVNs. Intelligent Peripheral (IP) was included in IN/1+ architecture and its function were further extended in IN/2. Vendor Feature Node (VFN) was also added to IN/2 architecture as one of the 6 essential elements.

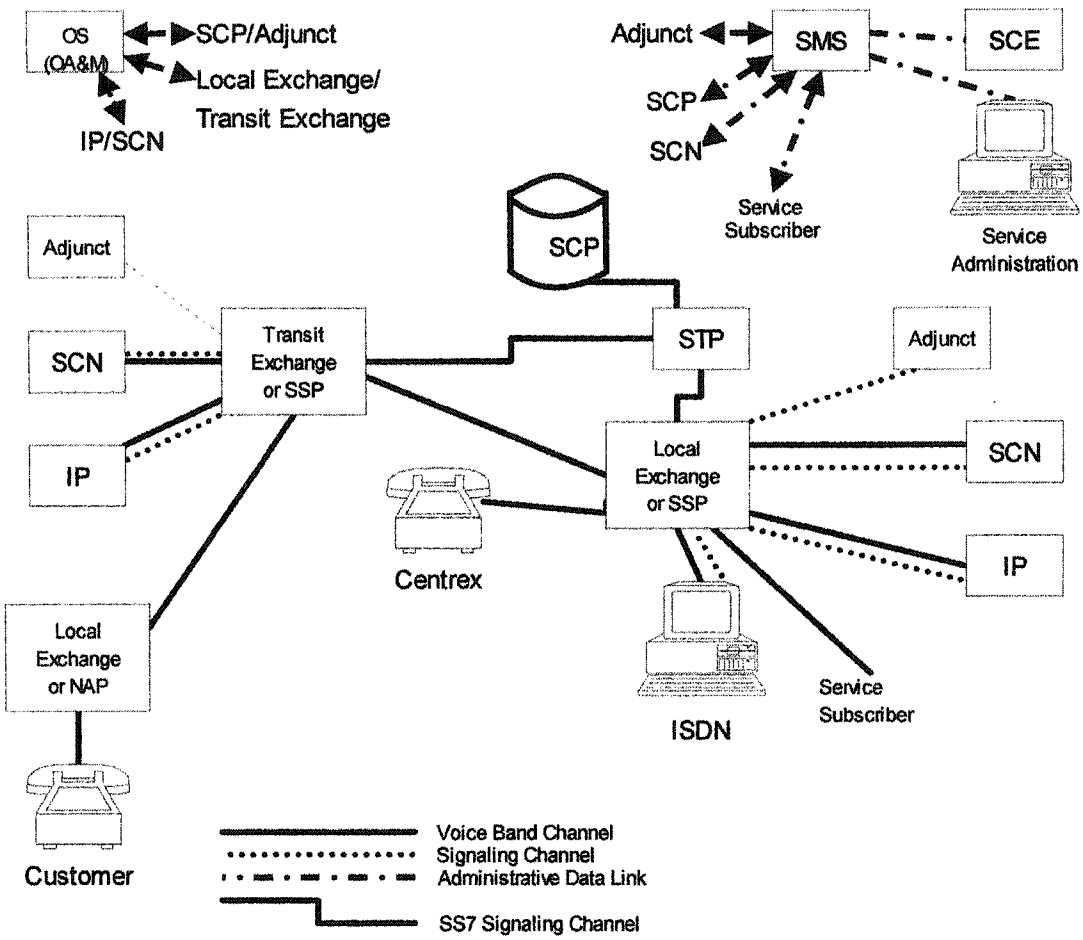
Evolution of IN is described slightly different in ITU-T 4-plane Intelligent Network Conceptual Model (INCM) [7]. There are Service Plane,

Global Function Plane, Distributed Service Plane, and Physical Plane. The INCM is mostly used by foreign countries.

1.3.1 Advanced Intelligent Networks (AINs)

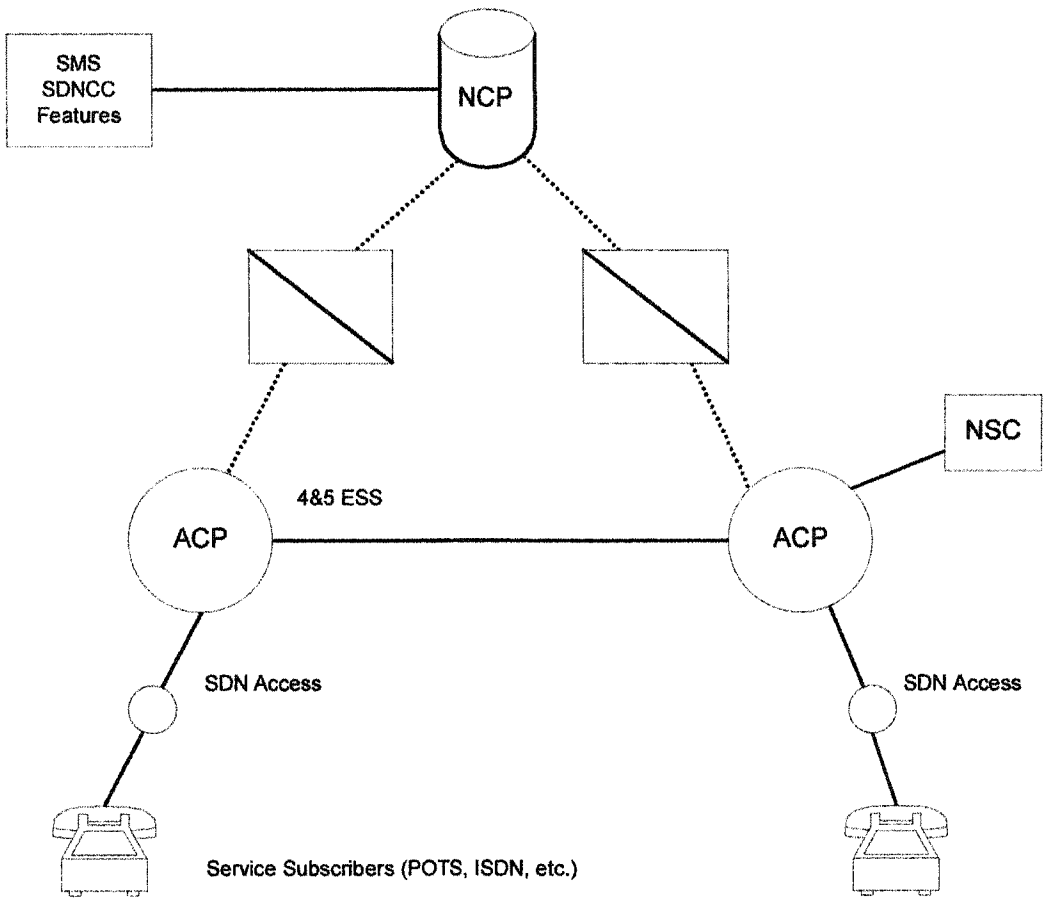
Figure 6 depicts an architecture of Advanced Intelligent Network [1]. PVN is a typical service provided by platform 0. IP can present a FIN accessed by circuit switch, packet switch, or ISDN.

Example of IP as Software Defined Network is shown on Figure 7. The virtual network defined by software commands permits private network owners, such as corporations, to use their own network in virtual cooperation with the switched public network [1].



OS = Operating System; IP = Intelligent Peripheral; SCN = Service Circuit Node; SCP = Service Control Point; SMS = Service Management System; SCE = Service Creation Environment; SSP = Service Switching Point; STP = Signal Transfer Point; NAP = Network Access Point; ISDN = Integrated Services Digital Network; OA&M = Operations, Administration and Maintenance.

Figure 6 Advanced Intelligent Network Platform 0.



SMS = Service Management System; SDNCC = Service Defined Network Control Center; NCP = Network Control Point; NSC = Network Services Complex; ACP = Access Control Point.

Figure 7 Software Defined Network.

AIN is just evolving into public domain intelligent networks. AINs can be readily implemented in private networks and tailored for business communication systems [1]. There are 5 basic components of AINs:

- Service Switching Points (SSPs).
- Service Control Points (SCPs).
- Signal Transfer Points (STPs).
- Intelligent Peripherals (IPs).
- Service Management Systems (SMSs).

These components perform a specific subset of functions necessary for most AINs. However the specific objectives of FIs dictate the architectural configurations that may or may not be suited in different telecommunications environments.

SSP receives and interprets the control signal (similar to interpreting the CPU operation code), inquires SCP via STP for information (like caller identification), performs logical channel switching (for example, completing a call on any given B or D customer channel).

SCP responds to queries from various SSPs. This information is crucial for completion of the requested service. Data bases can be resident in SCP or portable in AIN.

STP controls linkages to SSPs, SCPs, and IPs. In Public Domain INs (PDINs) STP is embedded in the SS7 network. IP can access Common Channel Signaling System via the SS7 protocol. For example, Transaction

Capabilities Application Part (TCAP) of SS7 can be utilized as STP for SSP to communicate with SCP.

IP performs voice and data services. The use of IP significantly depends on whether the AIN is specialized or generic. Specialized IP can have its own localized switching facility to contact additional telecommunication services, or private networks (intranets).

SMS provides administrative services, connects to SCP and Service Creation Environment (SCE). Communication link (normally X.25 in IN) is localized and distinct from AIN.

1.3.2 Financial Institution Networks

Private financial institution networks can be divided into external (extranet) and internal (intranet) networks. Bank's intranet connects worldwide bank's locations (branches) and headquarters. Bank's extranet connects different financial institutions (F2F). Large banks can act as a service bureau for smaller banks to connect them to the large bank's private extranet or to the shared by member-banks financial network (such as SWIFT).

1.4 Trends in Design of Financial Institution Networks

Financial Institutions Networks (FIN) evolve in phases:

- Value Added Networks (VAN), intranet in the 1980s,
- Private Virtual Networks (PVN), extranet in the 1990s and at present time,
- Intelligent Peripherals (IP) in the future (2000s).

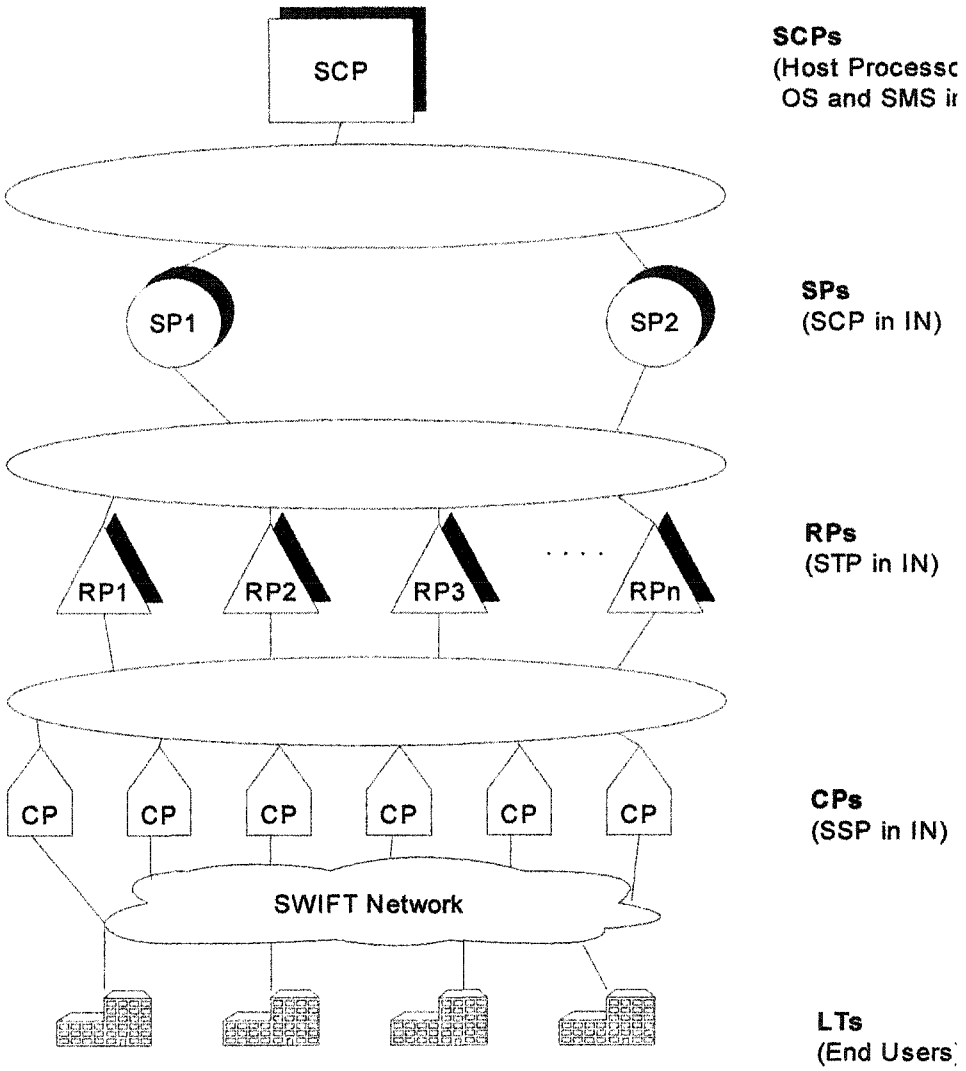
SWIFT is the more advanced PVN among the contemporary financial telecommunications networks shared by all banks in financial marketplace. The X.25-based SWIFT Transport Network (STN) architecture is presented on Figure 8 [20, 21].

In today's world FIN of a large bank (as PVN) can simulate almost all SWIFT features in communications with other financial institutions (F2F) bypassing SWIFT network.

Dramatic advances in internet technology, web design and standards (Java and XML), new challenges in e-market in recent years triggered the development of new generation of Financial Networks. SWIFT that dominates financial marketplace responded with new Secured IP-based Network (SIPN) and new network services INTERACT, SWIFTNet FIN and FILEACT. These services will be mandated to all financial institutions using SWIFT by the end of 2004 [21].

The System Control Processor (SCP) on Figure 8 is responsible for the operation of the entire SWIFT system. It constantly monitors and controls all

of the active system components as well as access to the system. There are 2 SCPs at the US control center and 2 SCPs at the Netherlands control center. At any given time only 1 SCP is active and in direct control of SWIFT. The other 3 SCPs are on constant standby and are being continuously updated with configuration data by the active SCP. The SCP itself does not process financial messages.



SWIFT Terminology:
SCP = System Control Processor; SP = Slice Processor;
RP = Regional Processor; CP = Communications Processor;
LT = Logical Terminal;
IN Terminology:
OS = Operating System; SMS = Service Management System;
SCP = Service Control Point; STP = Service Transfer Point;
SSP = Service Switching Point.

Figure 8 SWIFT Architecture.

The numbered sequence of events on the logical SWIFT message flow on Figure 9 is the following:

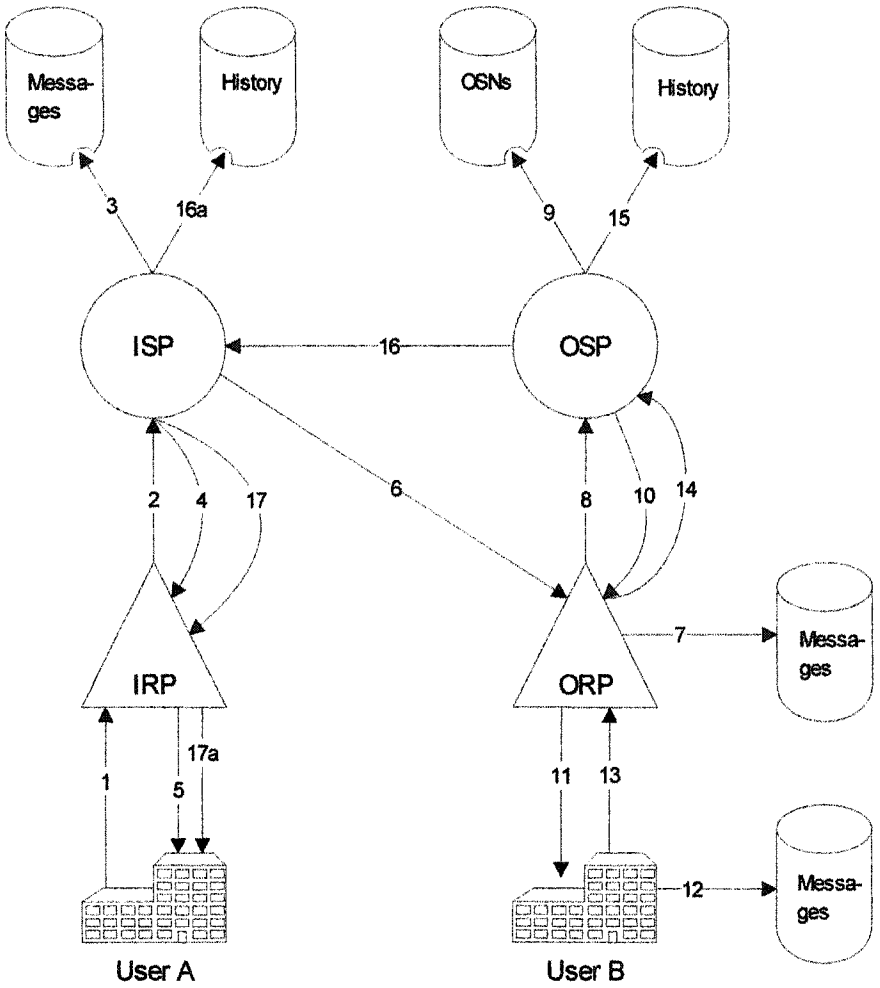
1. Having gained access to SWIFT, user A sends a message to Input Regional Processor (IRP), destined for user B.
2. IRP performs validation checks an the header, text and trailer of the message; checks that the Input Sequence Number (ISN) is correct and sends the message, along with Message Input Reference (MIR) and validation result to Input Slice Processor (ISP).
3. ISP safestores the incoming message to disk.
4. ISP sends a confirmation to IRP, signifying that the message, as received, is safely stored.
5. On receipt of this confirmation from ISP, IRP sends a positive Acknowledgement (ACK) or Negative Acknowledgement (NAK) to user A, giving notification of message acceptance or rejection. On receipt of an ACK, the user is assured that SWIFT has accepted responsibility for the delivery of that message. If a Negative Acknowledgement (NAK) is sent, this indicates that the message, although safestored (and hence retrievable), has not been accepted by SWIFT for delivery.
6. Having accepted a message, ISP determines (from its database similar to SCP in IN) which RP is the prime RP for user B and sends a copy of the message across the network to this RP (Output RP).

7. ORP temporarily stores the message on disk, and places it on one of the user-defined output queues for user B, where it is held awaiting delivery. The message will remain on hold until an LT at user B's destination has logged in for output and has asked to receive output messages from that particular output queue.
8. Before attempting to deliver the message, ORP assigns an Output Sequence Number (OSN) and creates a unique Message Output Reference (MOR) for that delivery attempt. ORP sends the MOR to user B's owning SP (OSP) and awaits authorization from OSP before attempting delivery.
9. OSP checks that the MOR assigned (and hence OSN) is valid for that particular LT, and records it in safestore.
10. OSP sends a confirmation to ORP, authorizing ORP to attempt to deliver the message using that MOR.
11. ORP outputs the message via the transport network to the appropriate LT, using the MOR authorized by OSP.
12. User B receives the output message via the appropriate LT and safestores it.
13. If the destination LT considers the message to have been properly received (i.e. checksum agree) a positive User Acknowledgement (UAK) is sent to ORP, confirming safe receipt. If the destination LT rejects the delivered message, a User Negative Acknowledgement (UNK) is returned to ORP and the message is considered to remain undelivered.
14. ORP creates a delivery history from the UAK/UNK and sends this to OSP.

15. OSP updates the message history with the result of this delivery attempt and records this in safestore.
16. OSP sends a copy of the message history to ISP for reconciliation, and ISP also safestores the message history (16a).
17. If Delivery Notification had been requested by user A, ISP (having received notification from OSP) sends a notification to IRP, which forwards the delivery notification to user A (17a). All messages sent may be retrieved online from message history queue (long term storage) for a period of up to 4 months from the day of input.

New challenges in financial industry dictate the trends in design of FINs:

- accumulated intelligence using adaptive infrastructure with new service creation allowing access to new generation financial networks;
- integration on base of single window concept for central access to variety of network services and network service applications;
- shared use by multiple financial institutions for the optimal utilization of resources, control, service provisioning, cost.



ISP = Input Slice Processor; OSP = Output Slice Processor;
IRP = Input Regional Processor; ORP = Output Regional Processor;
OSN = Output Sequence Number.

Figure 9 Logical SWIFT Message Flow.

Future financial networks (like SIPN) can replace other carriers including PVNs serving today's FINs such as FEDWIRE, CHIPS, EDI for transfer and routing of financial messages. B2B SWIFT's next generation approach is shown on Figure 10.

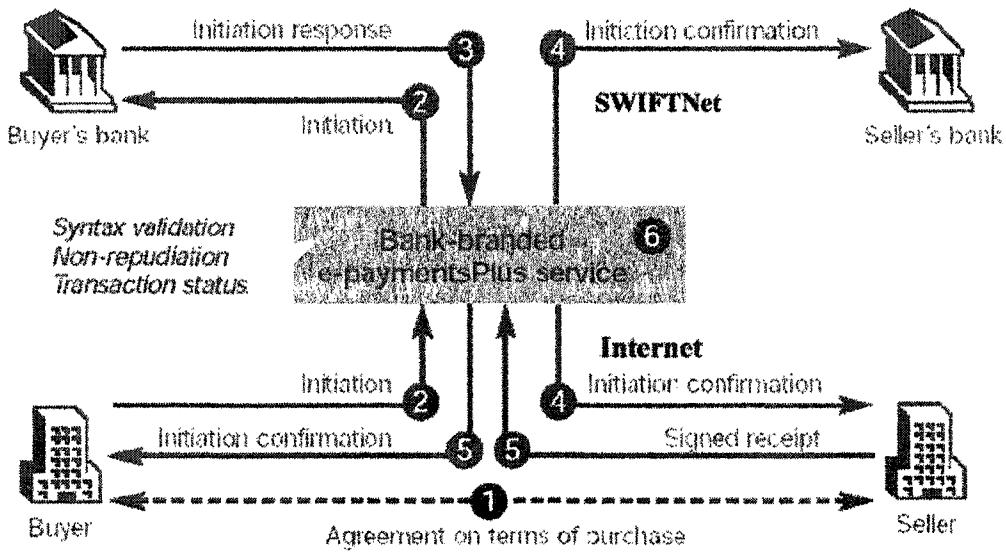


Figure 10 B2B SWIFT Next Generation Approach.

Sublayers of OSI application layer will be clearly defined for FIIN. FINs are being developed to make them truly intelligent and turn them into FIINs wich become IP components of AINs.

1.4.1 Accumulated Intelligence

The main characteristic of intelligence in a network is the ability to carry and communicate information with distinct algorithmic adaptation.

Adaptability occurs locally in a node (for ex., in residential programs) or globally between nodes for extraneous functions [1].

Financial networks carry system and financial messages (business transactions) where message is a logical unit of data. Thus financial message is a single unit of work carried over financial network. This unit of work is indivisible on upper 4 layers of OSI model: application, presentation, session, transport. Then splitting message into packets depends on underlying protocol: IP, ATM, X.25, Frame Relay.

Today's financial networks are message switching networks. Dedicated path is not established between sender and receiver (i.e. between correspondent banks) of a message. Both correspondents do not have to be available at the same time (different national holidays, time zones, etc.). Rather, each message has a header with source and destination addresses. A message is passed through he network from one node to another. At each node the entire message is received, stored, and then transmitted to the next node.

This type of network is called store-and-forward. If receiver is not available, the network is pending its availability to deliver the stored message. The waiting time depends on service agreement. SWIFT's default time, for example, is 7 days before it disconnects the correspondent.

Message size depends on the network used. SWIFT uses ISO 15022 since November 2001. These standards limit a message length to 10K bytes. TELEX carriers normally allow up to 32K-byte messages. Messages exceeding

the maximum length need to be fragmented by the sender. Each part of a multipart message in this case is considered by financial network as independent message. It has to be assembled by the receiver.

Message switching is not appropriate for interactive traffic. Message delay through the network is long. Delay varies highly from 1 sec to 10 min in average and increases with increased load. (Backlog can reach hours during end of month pick times).

Future financial networks are being developed based on packet switching technique: first of all IP, and to some extend Frame Relay, ATM, X.25. Each router transmits a packet as soon as it arrives. It does not wait for entire message. The packet is not stored. It might be kept temporarily for recovery until the packet is received by the next router (or its destination).

Maximum packet size depends on the protocol and might be negotiable within certain range: 512 bytes for IP, 48-byte payload for ATM, 128 bytes for X.25. Most carriers offer Frame Relay, ATM, X.25 services with the speed 1 Mbps, 156 Mbps, and 56 kbps correspondingly.

Packet switching is fast enough for interactive traffic.

Any FIN serves two main functions:

- Routing between bank's products (back-end applications) and a network..
- Connection to a network (extra- or intranets).

Construction of FIN as Intelligent Peripheral specialized for financial services with its own localized switching facility to perform these two functions becomes a challenge.

Each bank runs so called Bank Messaging System (BMS) which performs all or most FIN functions. Each financial message belongs to some bank's product, in other words to some back-end application. Operation areas for that application can be located at different bank's branches worldwide. Applications and locations are usually outside of the BMS. The basic message flow is as follows:

- "Outbound" message is originated in some application at certain location, delivered to BMS, routed to required financial network, and sent out of the bank. It will be delivered to its destination by a network.
- Acknowledgment is received by BMS, reconciled with original message and routed to an application at location originated a message.
- "Inbound" message is routed by BMS to required application at targeted location.
- Delivery acknowledgment is provided by BMS to a network from which a message was received.

Let denote the set of applications as $Y = \{y_1,...,y_j\}, j = \{1,...,m\}; m \geq 1$; the set of locations as $X = \{x_1,...x_i\}, i = \{1,...,n\}; n \geq 1$. Any subset of locations

$X_l \subset X, X_l = \{x_{i_1},...,x_{i_k}\}, i_k \in \{i\}, k \le n, l = \{1,...2^n - 1\}, (\sum_{k=1}^{n} \binom{n}{k} = 2^n - 1)$ can operate any subset of applications

$$Y_p \subset Y, Y_p = \{y_{j_1},...,y_{j_r}\}, j_r \in \{j\}, r \le m, p = 1,...2^m - 1, (\sum_{p=1}^{m} \binom{m}{p} = 2^m - 1).$$

All active locations C(n,k) and applications C(m,p) binary relationships $l * p = \{1,...,(2^n - 1) * (2^m - 1)\}$ are predefined in BMS at configuration time by parameter setting. Processing or routing of a message within BMS follows the business rules defined for that type of message. All standard routing of a message is encoded in BMS. This becomes an intelligent component of BMS infrastructure. Adoption of a new location x_{n+1} to the existing application or a new application y_{m+1} to the existing or new location becomes an easy task of setting new values for the existing parameters. No coding is required if there are no exceptions in the business rules. Exceptions need to be coded until they become standard. This constitutes accumulation of intelligence with infrastructure adopting more and more locations-applications over time.

Thus FIN contains SMS with embedded SCE that dramatically lower the cost of adopting the new services.

For example, in legacy system the introduction of a new application required hard-coding for each location-application combination. The full software development life cycle (SDLC) of an average complexity application

that included analysis, design, coding, unit and integrated testing, implementation normally took 10 person/weeks. In FIIN approach for the same application it takes 1 person/week. The added intelligence cut the development expenses by the order of magnitude (about 10 times). This is not counting savings for operation, monitoring and maintenance which are significantly simplified and streamlined. All those savings are highly appreciated by current users.

1.4.2 Integration

FIN connects to financial networks. This function is very similar to connection provided by ISP to the Internet. The difference is in multiplicity of financial network services (STN FIN, SIPN FIN, InterAct, FileAct, IC Telex, CHIPS, etc.), and financial network service applications (CLS, RTGS, etc.). The integrated platform becomes a necessity. The single window concept provides the following:

- Central access.
- Central security administration.
- Central configuration, monitoring, and operation.

Central access allows bank's products to connect via the central place to variety of financial networks including future financial networks of new generations.

Central security administration manages the following:

- User access and system privileges.
- PKI keys and certificates.
- Encryption and decryption services.

Central configuration, monitoring, and operation is responsible for gateways connecting to networks. All gateway are controlled locally or remotely using browser-based Graphic User Interface (GUI).

Single window concept using MQ Series as example is shown on Figure 11.

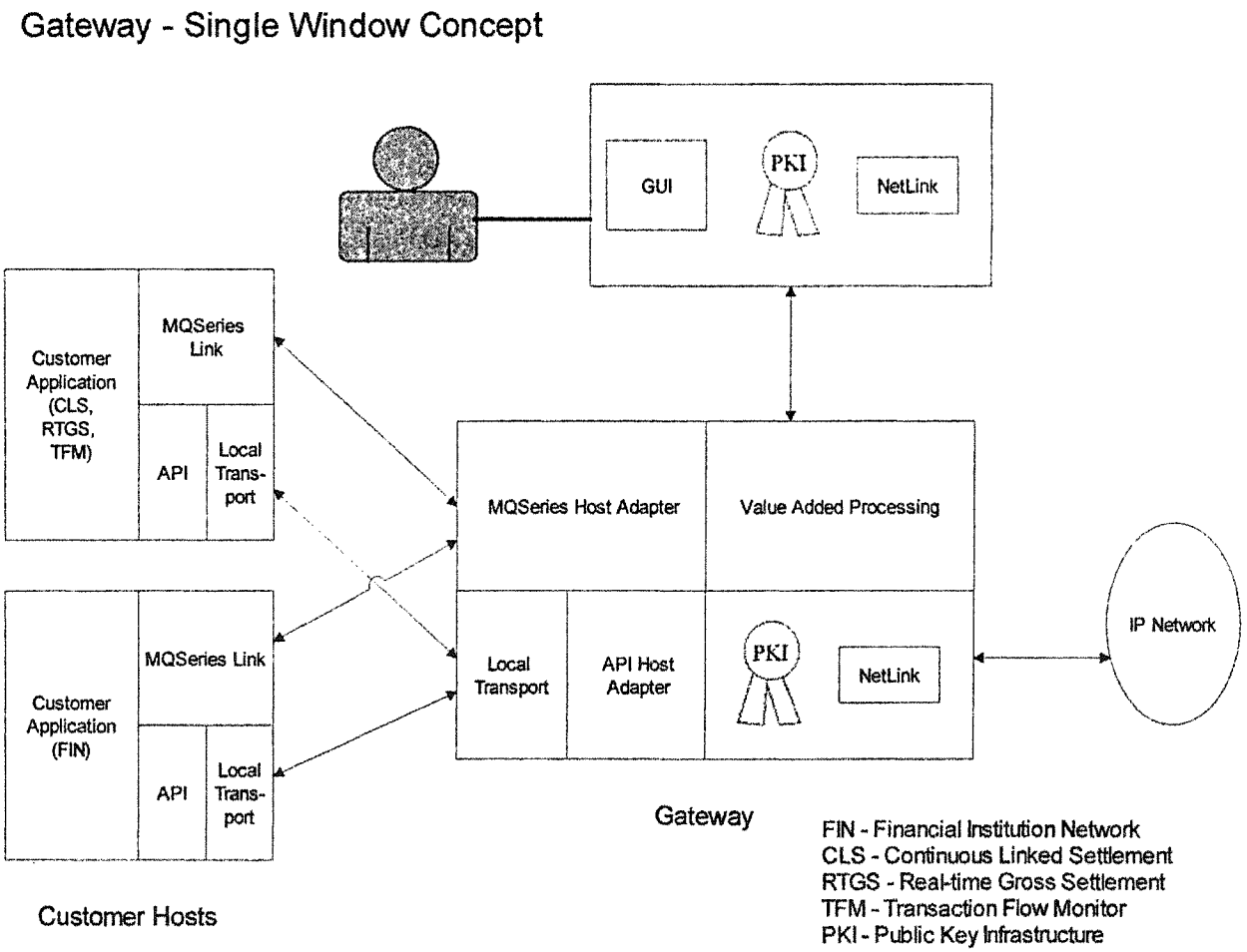


Figure 11 Single Window Concept.

1.4.3 Shared Use

Only large financial institution can afford to implement FIN with the whole spectrum of financial and network services. The intelligent component needs to be added to FIN to allow large FI act as service bureau for smaller FIs. It means that multiple FIs can share the same FIN. The challenge here is to separate processing and data of one FI from being accessed by other FIs without jeopardizing performance or any other functions of FIN.

The adaptive infrastructure with accumulating intelligence described earlier as method to add new applications/location combinations suites well for adoption of new FIs. That approach contributes greatly to completion of real FIIN.

The method of combining of any subset of participating locations with any subset of available applications is also applicable to other areas of business outside of financial industry (i.e. manufacturing, insurance, pharmaceutical, etc.).

2 Architecture of Financial Institution Intelligent Network (FIIN)

Large corporations have very complicated worldwide intranets connecting their LANs and WANs internally and externally to the Internet and other specialized and private Value Added Networks (VAN). Financial Institutions use SWIFT, Government Reserve Banks, Clearing Houses' Networks, Public Carriers for TLX/FAX.

2.1 Hypothetical Model Functional Requirements

Hypothetical model represents the commonality of functions performed by variety of FIs and helps to define the requirements and design the architecture of the next generation of FI messaging systems.

For instance, the common functions performed by New York FIs are the following:

- Send and receive all U.S. domestic payments to the New York Federal Reserve Bank and the New York Clearing House
- Send and receive securities instructions to the New York Federal Reserve itself, and as a conduit to the Federal Reserves around the country.

- Send and receive worldwide transactions (messages) over the SWIFT network.
- Send and receive messages via TELEX.
- Send and receive messages via FAX.

The reference architecture for financial messaging identifies the key software components, their functional roles and interfaces. This architectural framework and the component definitions can also be used to develop software products that might fill the different architectural roles.

The key functional features of the existing FI messaging systems will be preserved in the near future developments. In this section the commonalities in the message flow of each sub-system are highlighted as well as new beneficial features are described. The next generation financial messaging software will facilitate the commonalities of message processing without sacrificing existing functionality.

The list of functional requirements of hypothetical model was not intended to be exhaustive. The functional features of existing financial messaging systems were studied to emphasize the commonality and applicability to the new strategic initiatives in the financial industry. One example of such an initiative is IBM WebSphere Business Integrator for Financial Networks (see [20]. Each requirement or feature might have its own intricacies that are beyond the scope of this work. .Any actual deployment will require the detailed elaboration of specific requirements for a particular FI.

2.1.1 Basic Structure of Queuing Model

The basic queuing process was used in the study (see [24]). Bank Messaging System {BMS} of any FI is a messaging network of queuing systems or queuing network. The assumptions for the typical queuing model (see Figure 12) are the following:

Input source size is unlimited, i.e. the total number of messages that might require service from time to time is relatively large finite number. In other words, the rate at which the input source generates new messages is not affected by the number of messages being already in the queuing system.

A queue is infinite. Messages are waiting in a queue before being served. Max queue depth is depicted with the relatively large finite upper bound on the permissible number of messages (usually 50,000). An exceptional case of balking is when messages overflow the max and fall into so-called "dead letter queue" from which they have to be later recovered. Those cases are monitored and personal (help desk) is alarmed for exceptions in normal processing.

Queue discipline is FIFO. The complicating factor is commit point.

Service mechanism has one service facility with a finite number of parallel servers (channels). Same distributions for all servers. Normally one server per queuing system (i.e. per one queue).

If commit point is *m* messages, then after the *m*-th message is served all *m* messages are removed from the queue and service considered completed. Thus service time includes all messages until commit point is reached.

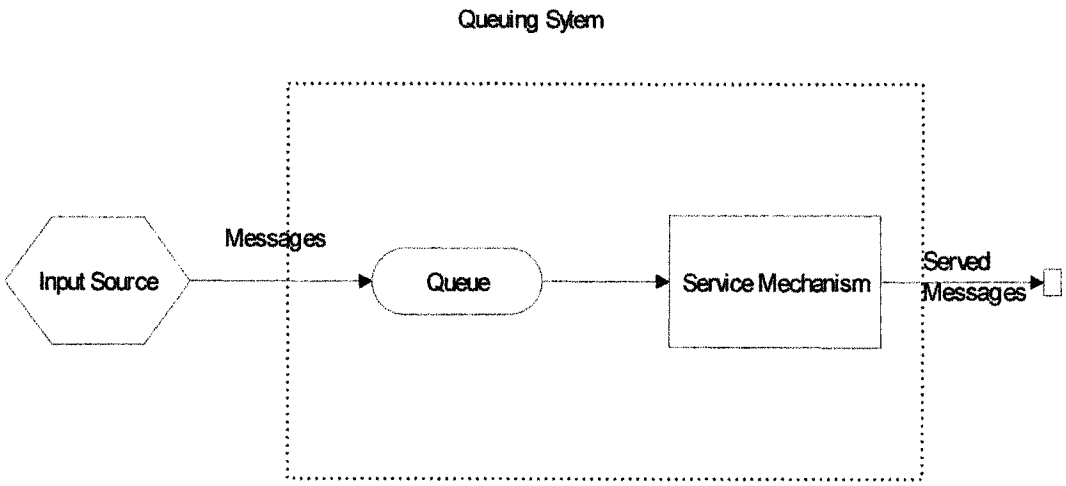


Figure 12 Basic Queuing Process

This study uses standard queuing theory terminology and notation as follows:

State of system = number of messages in a queuing system.

Queue length = number of messages waiting for service to begin or

= state of system minus number of messages being served.

$N(t)$ = number of messaged in queuing system at time t ($t \geq 0$).

$P_n(t)$ = probability of exactly n messages in queuing system at time t, given number at time 0.

s = number of servers (parallel tasks) in queuing system.

λ_n = mean arrival rate (expected number of arrivals per unit time) of new messages when n messages are in system.

μ_n = mean service rate for system (expected number of messages completing service per unit time) when n messages are in system.

The following assumptions are appropriate for BMS:

$\lambda_n = \lambda \ \forall n$, mean arrival rate is independent on a number of messages being in system. Also because λ_n is the mean arrival rate while the system in state n (n=0,1,2,...) and P_n is the proportion of time when the system is in state n, then $\lambda = \sum_{n=0}^{\infty} \lambda_n P_n$.

$\mu_n = \mu \ \forall n \geq 1$ and s=1, mean service rate is a constant per only busy server.

$\mu_n = s\mu \ \forall n \geq s$, mean service rate is a constant when all s servers are busy.

Then the following is true:

$1/\lambda$ = expected inter-arrival time.

$1/\mu$ = expected service time.

$\rho = \lambda/(s\mu)$ = utilization factor for the service facility, i.e. the fraction of the system's service capacity ($s\mu$) that is being utilized on the average by arriving messages (λ).

If $\rho \geq 1$, the state of the system will grow and can lead to the system outage over time (during bursts of traffic). This situation has to be monitored.

When system begins operation (after weekly maintenance) the state of the system depends on initial state and elapsed time (so-called transient condition of the system). The system is said to be in **steady-state condition** when it becomes essentially independent of the initial state and the elapsed time. The probability distribution of the state of the system remains the same (stationary distribution) over time. The following notation is used for the system in steady-state condition:

P_n = probability of exactly n messages in queuing system.

L = expected number of messages in queuing system, $L=\sum_{n=0}^{\infty} nP_n$.

L_q = expected queue length (excludes messages being served),

representation of how the status of the physical (say messaging) system evolves over time. Time can be discrete or continuous.

A continuous time stochastic process $\{X(t'); t' \geq 0\}$ is a continuous time Markov chain if it has the Markovian property which in turn can be defined as follows:

$P\{X(t+s)=j|X(s)=i \text{ and } X(r)=x(r)\} = P\{X(t+s)=j|X(s)=i\}$,

$\forall i,j=0, 1, \ldots, M$ and $\forall r \geq 0$, $s>r$, and $t>0$, where

$t'=r$ is the past time,

$t'=s$ is the current time,

$t'=s+t$ is t time units into the future.

$P\{X(t+s)=j|X(s)=i \text{ and } X(r)=x(r)\}$ is a transition probability. We will consider only stationary transition probabilities which are **independent of s**, such that $P\{X(t+s)=j|X(s)=i\} = P\{X(t)=j|X(0)=i\}$ $\forall s>0$. Another notation: $p_{ij}(t) = P\{X(t)=j|X(0)=i\}$ where $p_{ij}(t)$ is continuous time transition probability function and $\lim_{t\to 0} p_{ij}(t)=1$ if $i=j$, $\lim_{t\to 0} p_{ij}(t)=0$ if $i\neq j$.

The most important probability distribution in queuing theory is the **exponential distribution** that can be expressed as follows:

$f_T(t)=\alpha e^{-\alpha t}$ $\forall t \geq 0$ and $f_T(t)=0$ for $t<0$ where T is inter-arrival or service (event) time with parameter α. This function is strictly decreasing $\forall t \geq 0$. Let X(t) be the number of events by time t ($t \geq 0$). Then X(t) has a **Poisson distribution** if

$$L_q = \sum_{n=1}^{\infty}(n-s)P_n$$

W = waiting time in system (includes service time) for each individual message.

W_q = waiting time in queue (excludes service time) for each individual message.

The following have been proven (see [24]):

$L=\lambda W$, $L_q = \lambda W_q$, $W = W_q + 1/\mu$. This relationship was used in the study.

Note that system can reach the steady-state conditions only and only if $\rho = \lambda/(s\mu)<1$

The so-called **birth-and-death** process is very important in the queuing theory. The applicability of the birth-and-death process to the messaging system is worth researching. This is a special type of **continuous time Markov chain**.

A stochastic process is an indexed collection of random variables $\{X_t\}$, where the index t runs through a given set T. Stochastic process describes behavior of a system operating over some period of time T. The random variable X_t represents the current status or the state of the system at time t. The system is observed at particular points of time, say t=0,1,2,... . Thus the stochastic process $\{X_t\}=\{X_0,X_1,X_2,\ldots\}$ provides a mathematical

$$P\{X(t)=n\}=\frac{(\alpha t)^n e^{-\alpha t}}{n!}, \text{ for } n=0,1,2,\ldots .$$

For n=0 $P\{X(t)=0\}=e^{-\alpha t}$ which is the probability that the first event occurs after time t.

The mean of this Poisson distribution is $E\{X(t)\}= \alpha t$, so that the expected number of events per unit time is α. In other words, α is the mean rate at which the events (arrival or service) occur. The event counting process on a continuing basis is called a **Poisson process** with parameter α. In application to arrivals with mean arrival rate $\alpha=\lambda$ it is called **Poisson input process** with parameter λ. The queuing model in this case has a **Poisson input**.

Balance equation expresses a key principle in queuing theory:

Rate In = Rate Out

In other words, for any state of the system n (n=0,1,2,...) mean entering (arrival) rate = mean leaving (service) rate.

The assumptions for the birth-and-death model are the following:

The arrival (birth) probability distribution is Poisson input with parameter λ_n.

The service completion (death) probability distribution is Poisson exponential distribution with parameter μ_n.

$\mu_1 P_1 = \lambda_0 P_0$, … , $\lambda_{n-1} P_{n-1} + \mu_{n+1} P_{n+1} = (\lambda_n + \mu_n) P_n$ for n=0,1,2,… .

Let define C_n = 1 for n=0 and denote $C_n = \frac{\lambda_{n-1}\lambda_{n-2}\cdots\lambda_0}{\mu_n \mu_{n-1}\cdots\mu_1}$, for n=1,2,…Then the steady-state probabilities are : $P_n = C_n P_0$, for n=0,1,2,… .

Given $\sum_{n=0}^{\infty} P_n = 1 \Rightarrow (\sum_{n=0}^{\infty} C_n) P_0 = 1 \Rightarrow P_0 = (\sum_{n=0}^{\infty} C_n)^{-1}$.

The performance of a queuing system is measured by the following key parameters: L, L_q, W, W_q. It can be obtained after calculating P_n.

Overall formulas based on balance equation and birth-and-death process are presented in Table 1.

$P_n(t)$ = probability of exactly n messages in queuing system at time t, given number at time 0.
s = number of servers (parallel tasks) in queuing system.
λ_n = mean arrival rate (expected number of arrivals per unit time) of new messages when n messages are in system.
μ_n = mean service rate for system (expected number of messages completing service per unit time) when n messages are in system.
λ_n = λ $\forall$n, mean arrival rate is independent on a number of messages being in system.

μ_n = $s\mu$ $\forall n \geq s$, mean service rate is a constant when all s servers are busy.

$1/\lambda$ = expected inter-arrival time.

$1/\mu$ = expected service time.

$\rho = \lambda/(s\mu)$ = utilization factor for the service facility, i.e. the fraction of the system's service capacity ($s\mu$) that is being utilized on the average by arriving messages (λ).

L = expected number of messages in queuing system.

L_q = expected queue length (excludes messages being served).

W = waiting time in system (includes service time) for each individual message.

W_q = waiting time in queue (excludes service time) for each individual message.

$C_n = 1$ for n=0 and $C_n = \frac{\lambda_{n-1}\lambda_{n-2}\cdots\lambda_0}{\mu_n\mu_{n-1}\cdots\mu_1}$, for n=1,2,...

$P_n = C_n P_0$, for n=0,1,2,... $P_0 = (\sum_{n=0}^{\infty} C_n)^{-1}$

$\rho = \lambda/(s\mu) < 1$ $\lambda = \sum_{n=0}^{\infty} \lambda_n P_n$

$L = \sum_{n=0}^{\infty} nP_n$ $L_q = \sum_{n=1}^{\infty} (n-s)P_n$

$W = L/\lambda$ $W_q = L_q/\lambda$

Table 1 Performance Formulas for Exponential Queuing Model

The most typical and widely used model in queuing theory is M/M/s, i.e. inter-arrival times are independently and identically distributed according to Poisson input, service times are independently and identically distributed according to another exponential distribution, and the number of servers is s. The results for single and multiple servers can be found in [24] and presented in Table 2.

For M/M/s Queuing Model:

$$L=\frac{\lambda}{\mu-\lambda},\ L_q=\frac{\lambda^2}{\mu(\mu-\lambda)},\ W=\frac{1}{\mu-\lambda},\ W_q=\frac{\lambda}{\mu(\mu-\lambda)} \quad \text{for } s=1$$

$$L=L_q+\frac{\lambda}{\mu},\ L_q=\frac{P_0(\lambda/\mu)^s\rho}{s!(1-\rho)^2},\ W=W_q+\frac{1}{\mu},\ W_q=\frac{L_q}{\lambda} \quad \text{for } s>1$$

Table 2 Performance Measurements for M/M/s Queuing Model

The exponential distribution is not applicable when inter-arrival or service times are scheduled or regulated. Some results are available for queuing models involving non-exponential distributions (see Table 3).

Queuing model is used to determine most efficient way to operate a queuing system. Excessive service capacity is costly. Lack of service capacity results in excessive waiting backlogs, and missing SLAs (Service Level Agreements). Finding an appropriate feasible balance between the service

capacity (hardware platform, number of CPUs, etc.) and waiting time is a challenge.

For **M/G/1** Queuing Model with Poisson input and independent service times with the same $1/\mu$ mean probability distribution and variance σ^2:

$P_0=1-\rho$, $L=\rho+L_q$, $L_q=\frac{\lambda^2\sigma^2+\rho^2}{2(1-\rho)}$, $W=W_q+\frac{1}{\mu}$, $W_q=\frac{L_q}{\lambda}$.

For **M/D/1** Queuing Model with degenerate (constant) service times where $\sigma^2=0$ $L_q=\frac{\rho^2}{2(1-\rho)}$.

For **M/E$_k$/1** Queuing Model the Erlang service time distribution is as follows:

$f(t)=\frac{(\mu k)^k}{(k-1)!}t^{k-1}e^{-k\mu t}$, for $t\geq 0$, $\mu>0$, $k=1,2,\ldots$- shape parameter.

Mean service rate is μ, variance $\sigma^2=1/(k\mu^2)$.

$L=\lambda W$, $L_q=\frac{1+k}{2k}\frac{\lambda^2}{\mu(\mu-\lambda)}$, $W=W_q+\frac{1}{\mu}$, $W_q=\frac{1+k}{2k}\frac{\lambda}{\mu(\mu-\lambda)}$.

Table 3 Performance Measurements for Non-exponential Queuing Models

The queuing theory **model** representing the real system has to be sufficiently realistic to provide reasonable predictions. The assumed form of the probability distribution of inter-arrival and service times should be sufficiently simple such that the model is mathematically tractable.

2.1.2 Queuing Model Selection

Message arrivals of the real BMS are occurring randomly. They appear on the time line as being clustered around the daily "dead lines" with occasional large gaps separating clusters. There is substantial probability of small inter-arrival times and the small probability of large inter-arrival times. The next arrival does not depend on when the last arrival occurred. In other words, every time period of fixed length has the **same** chance of having an arrival regardless of when the preceding arrival occurred. The "dead lines" are regulated or scheduled.

Each message requires essentially identical service, i.e. the server performs the same sequence of service for a message with the expected service time. Small deviations occur due to the minor efficiency deviations of the server. Network ready queues have different service pattern, i.e. server might wait until application window becomes available.

The exponential probability distribution

$f_T(t)=\alpha e^{-\alpha t}$ $\forall t\geq 0$ and $f_T(t)=0$ for $t<0$ where T is inter-arrival or service time with parameter α is strictly decreasing function and therefore not realistic for probability distribution of inter-arrival and service times of the real BMS even though some properties can be helpful.

Let **event** be a completion of arrival or service. Arrival is completed when a message generated by some input source is written (stored) on a queue

and ready for service. Service is completed when a message is read, routed out of a queuing system, and removed from a queue (and therefore from a queuing system). Remember that a queuing system by definition contains only one queue (see Figure 12).

Service complicating factors are the followings:

Server can route a message to none (dumping a message) or multiple destinations, namely: another queuing system, IP-addressed device (printer, fax machine, PC), communication line, other targets.

Server removes a message from a queue (completes service) only after a predefined commit point is reached. If commit point is $m>0$ messages, then all m messages are removed only after each immediately receiving system confirms an acceptance of its message. Otherwise the service process is rolled back for all m messages and restarted after problem is solved.

This research concludes that the most appropriate BMS queuing model can be labeled as GI/G/1 (general labeling is described in [24]), where:

GI=general independent distribution of inter-arrival times (en-queue rate). Normally clustered (bursts of traffic) before business dead line times during a business day, start and end of business day (SOB, EOB), end of week, month, quarter, year.

G=general distribution (any arbitrary distribution) of service times (de-queue rate or rate out). No restriction on distribution of service times (limited by hardware capacity).

1=number of servers are restricted to be exactly one. Running more than one server (parallel tasks) against one queue exposes to non-recoverable situation from a commit (synchronization) point.

According to [24]: "The mathematical analysis of queuing models with non-exponential distribution is much more difficult" and not widely available. Performance was measured empirically for a number of queuing systems within BMS and the results for one queue corresponding to GI/G/1 model are shown on Figure 13. This graph does not represent neither exponential nor degenerated distribution of arrival or service times.

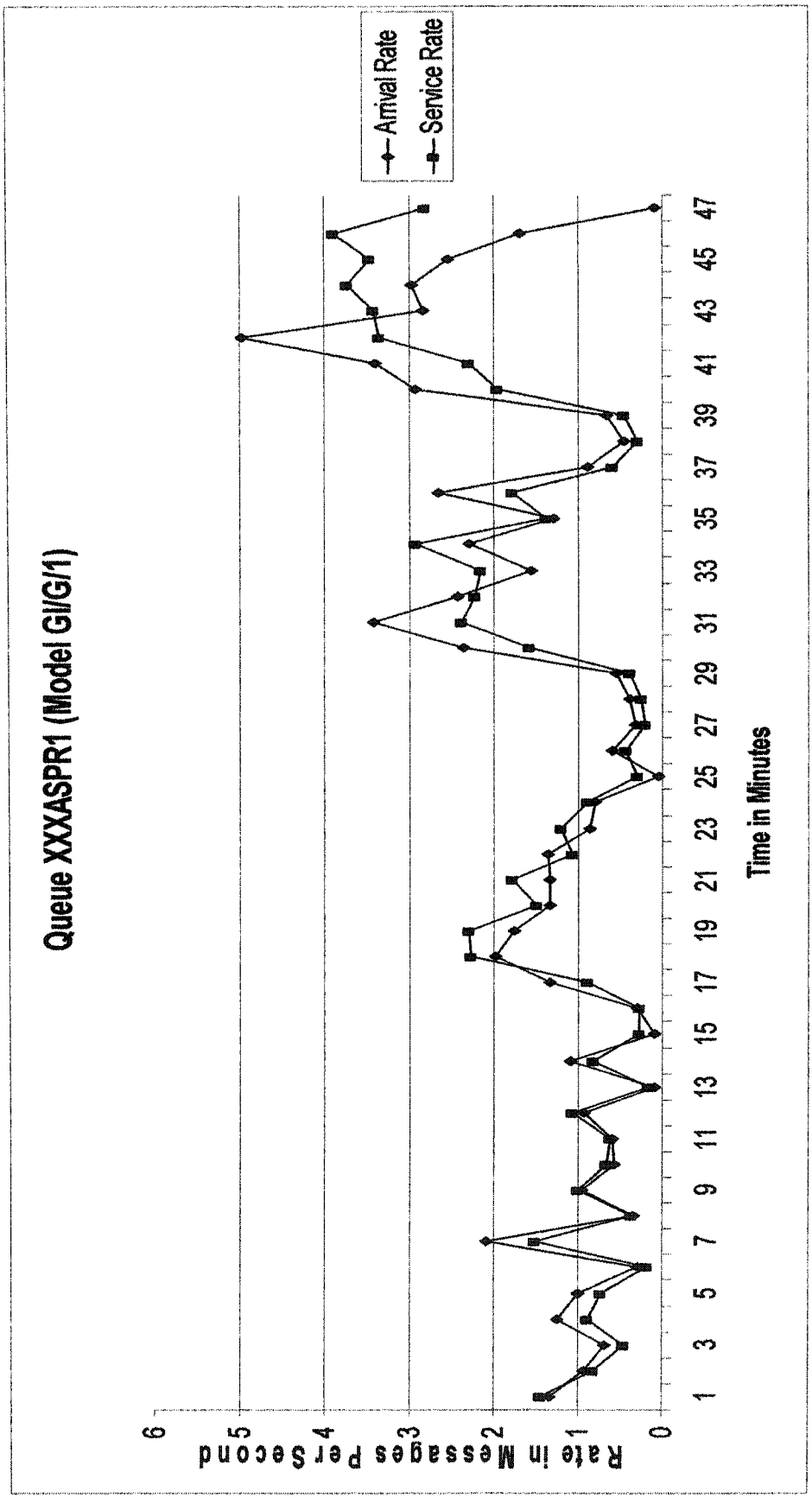


Figure 13 Queue Performance Measurement for GI/G/1 Model.

The real BMS instance contains network of queuing systems (**queuing network**). Therefore it was necessary to study the entire network to obtain such information as the expected state of the entire system (number of messages in the system), expected waiting and service time. The queuing discipline was non-preemptive priorities between individual queuing systems even thought it was single-threaded FIFO discipline within each queuing system. That is if message A is retrieved for service by a single server at the time when message B arrived in the higher priority queuing system, the service for message A is completed (message A is not returned back to the queue) before message B is served. Servers for each queuing system can also be run in parallel up to the predefined number of tasks.

For the entire queuing network the balance equation principle has to be preserved to avoid exceptionally long wait or system crash. In other words, for any state of the queuing network n (n = 0, 1, 2, ...):

mean arrival (entering) rate = mean service (leaving) rate or

Rate In = Rate Out

One BMS instance of a large FI contains about 5,000 queues i.e. messaging network of 5,000 queuing systems where the queuing process is applicable to each queue. The BMS monitoring system was developed to

measure arrival/service (en-queue/de-queue) load every minute. The queuing network load sample representing the same time when queue rate was taken is shown on Figure 14. The graph trend holds throughout a business day as well as formula above (rate in = rate out). This is specifically the case because the business hours are staggered throughout the 24 hours of practically any calendar day around the world.

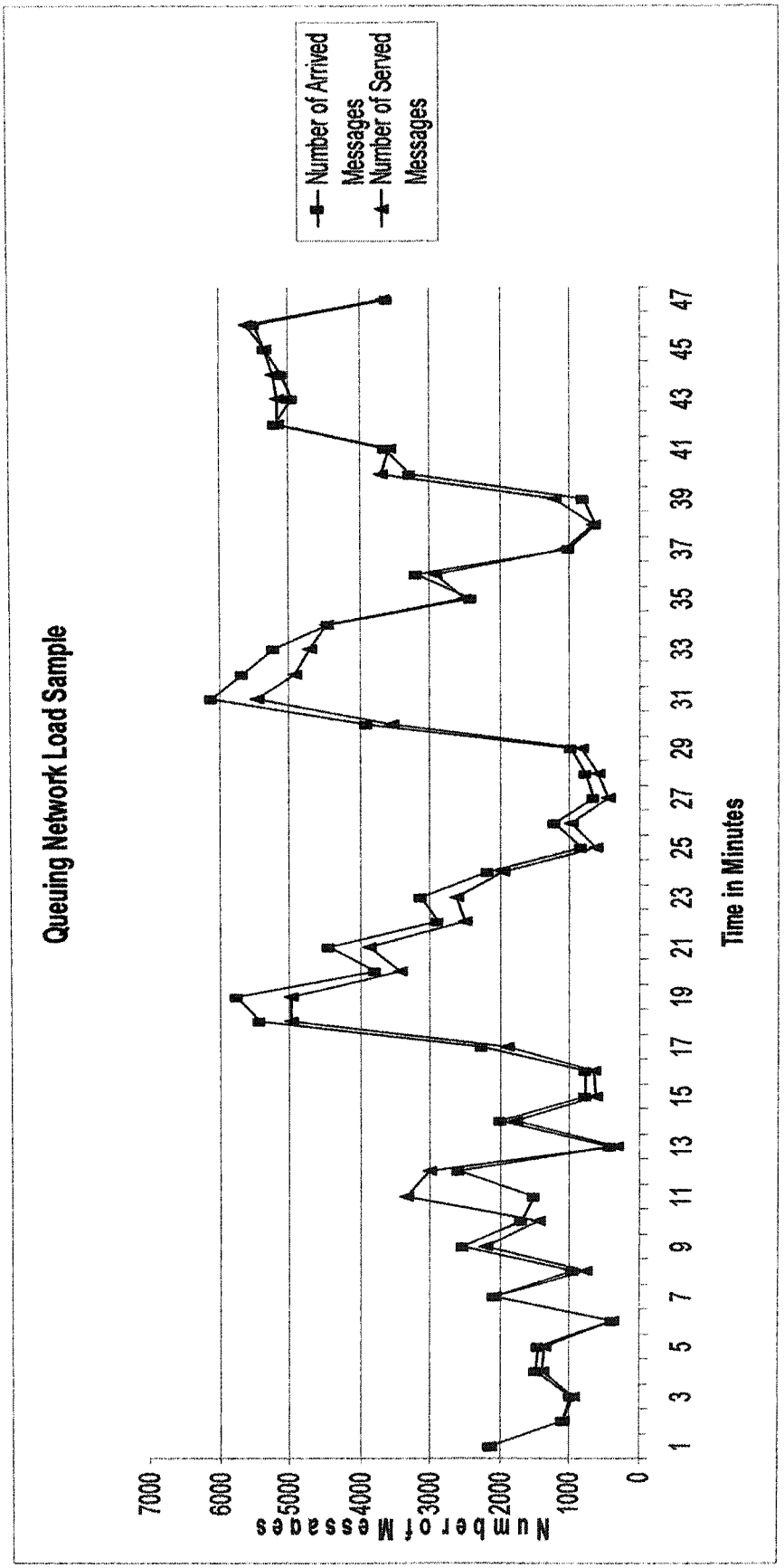


Figure 14 BMS Queuing Network Load Sample

2.1.3 Given Financial Messaging Requirements

Financial messaging system of any given FI is assumed to perform the following functions:

- Provide a reliable and fully recoverable means of exchanging messages with FI back-end systems.
- Provide the ability to route messages in all of the following possible ways:

  -From back-end systems to external destination(s)

  -From one back-end system to another, particularly for inter-branch traffic

  -From an external destination to one or more back-end systems or printers.
- Provide validation, enrichment, and transformation services to ensure that messages will be successfully processed at the ultimate destination
- Provide a reliable and fully recoverable means of exchanging messages with external parties via SWIFT, CHIPS, FEDWIRE, TELEX, FAX, or other proprietary point-to-point interfaces.

Because of the critical business role fulfilled by the financial messaging infrastructure, all components that make up the messaging infrastructure must provide robust features for monitoring, control, diagnostics, redundancy, and audit trail. Moreover, due to the enormous volume of messages that are currently being processed by FI messaging systems and the projections for

growth, it is critical that all components meet the highest standards for throughput and scalability.

A fundamental principle of a financial messaging infrastructure is that back-end applications should be loosely coupled with the messaging infrastructure, to insulate the back-end application as much as possible from changes imposed by external interfaces. In practical terms, this means that the syntax of messages sent and received by back-end applications should be specified by those applications, so that back-end application developers do not have to master the details of external interface specifications. The messaging infrastructure should take responsibility for transforming these messages to and from the formats defined in those specifications.

2.1.4 Performance and Capacity Planning

The following table shows daily and critical performance measurements, message volumes assumptions. It is assumed the volume growth will remain the same. The number of directly linked customers will increase.

Projected volume growth for a large FI over the five years:

Fedwire Funds, Chips Funds and Fedwire Securities traffic will double,

SWIFT, Telex, Fax and directly linked traffic will increase by 5 times.

A 100% safety margin has been added to the five-year projections to ensure throughput and performance.

The following table shows an example of daily volumes for a large FI and 5 year projected peak throughput.

	Daily AveragePeak Throughput			
Method of Payment	Inbound	Outbound	5 Year Projected Inbound	5 Year Projected Outbound
FEDWIRE FUNDS	25,000	30,000	100,000	120,000
CHIPS FUNDS	80,000	25,000	320,000	100,000
FEDWIRE SECURITIES	70,000	80,000	280,000	320,000
SWIFT (International)	80,000	90,000	480,000	540,000
SWIFT (Domestic)	160,000	160,000	960,000	960,000
TELEX	500	500	6,000	6,000
FAX	200	3,000	3,000	30,000
CPU LINKS (Point-to-point)	150,000	8,000	900,000	100,000

The following table shows the example of daily hourly peak and the 5 year projected throughput.

	Transaction Performance Criteria			
Method of Payment	Inbound Hourly Peak / Msg's per second	Outbound Hourly Peak / Msg's per second	Required Inbound Msg's per second	Required Outbound Msg's per second
FEDWIRE FUNDS	3,000 1 per second	4,000 1 per second	12,000 4 per second	15,000 5 per second
CHIPS FUNDS	25,000 7 per second	10,000 3 per second	100,000 29 per second	45,000 12 per second
FEDWIRE SECURITIES	15,000 4 per second	15,000 5 per second	60,000 17 per second	70,000 20 per second
SWIFT	60,000 16 per second	60,000 16 per second	750,000 210 per second	750,000 210 per second
TELEX	N/A	N/A	N/A	N/A
FAX	N/A	N/A	N/A	N/A
CPU LINKS (Point-to-point)	7,000 2 per second	N/A	130,000 36 per second	N/A

Latency is the time measurement from the receipt of the message to the completion of processing for that message so that it is ready to send to the recipient system. This does not include any waiting time due to the receiving system not accepting the messages as fast as they are being sent. Current FI

SWIFT systems can process messages faster then the SWIFT FIN system can accept. It is expected this bottleneck will continue to be the case.

The following table represents latency requirements

Method of Payment	Inbound (min)	Outbound (min)
FEDWIRE FUNDS	10	10
CHIPS FUNDS	10	10
FEDWIRE SECURITIES	20	20
SWIFT (International)	10	10
SWIFT (Domestic)	10	10
TELEX	30	30
FAX	30	30
CPU LINKS (Point-to-point)	10	10

2.1.4.1 Dead Lines.

Hourly peak rates normally occur around the hour of:

- Fedwire Securities Inbound 9:00 AM
- Fedwire Securities Outbound 9:00 AM & 11:00 AM
- Chips Funds Outbound 11:00 AM
- Fedwire Funds Outbound 11:00 AM
- Fedwire Funds Inbound 3:00 PM
- Chips Funds Inbound 3:00 PM
- Domestic Swift Inbound 3:00 PM
- Domestic Swift Outbound 3:00 PM

- International Swift Inbound 10:00 PM
- International Swift Outbound 10:00 PM

The following table shows current daily peak hour and the five-year projected throughput requirements. It is assumed one third of the day's traffic is processed in a peak hour, rounded to the nearest thousand.

	Message Throughput Requirements			
Method of Payment	Inbound Hourly Peak / Msg's per second	Outbound Hourly Peak / Msg's per second	Projected Inbound Msg's per second	Projected Outbound Msg's per second
FEDWIRE FUNDS	10,000 2 per second	12,000 3 per second	30,000 9 per second	40,000 12 per second
CHIPS FUNDS	25,000 7 per second	8,000 2 per second	100,000 30 per second	30,000 9 per second
FEDWIRE SECURITIES	20,000 6 per second	25,000 7 per second	90,000 25 per second	100,000 30 per second
SWIFT (International)	25,000 7 per second	30,000 8 per second	1,500,000 444 per second	1,800,000 500 per second
SWIFT (Domestic)	50,000 15 per second	50,000 15 per second	600,000 178 per second	600,000 178 per second
TELEX	150 1 per second	150 1 per second	2,000 1 per second	2,000 1 per second
FAX	N/A	800 1 per second	N/A	10,000 3 per second
CPU LINKS (Point-to-point)	3,000 2 per second	3,000 2 per second	60,000 18 per second	60,000 18 per second

Systems that process Fedwire Funds, Chips Funds, Fedwire Securities, Domestic SWIFT and International SWIFT generate a number of reports (assume 50 reports per day throughout the day). Report production is not critical performance measure.

All systems have to be fully recovered in 30 minutes.

The following assumptions were made for capacity planning:

	Summary of Projected Peak Throughput			
Method of Payment	Projected Inbound Daily Peak	Projected Outbound Daily Peak	Projected Inbound Hourly Peak	Projected Outbound Hourly Peak
FEDWIRE FUNDS	100,000	100,000	30,000	40,000
CHIPS FUNDS	300,000	100,000	100,000	40,000
FEDWIRE SECURITIES	300,000	300,000	90,000	100,000
SWIFT (International)	4,000,000	5,000,000	1,000,000	1,800,000
SWIFT (Domestic)	2,000,000	2,000,000	650,000	650,000
TELEX	6,000	6,000	2,000	2,000
FAX	200	30,000	20	10,000
CPU LINKS (Point-to-point)	200,000	200,000	60,000	60,000
TOTAL MSG's	6,706,200	7,736,000	1,932,000	2,702,000

Consider Average Message size is 1000 bytes, number of transformations per message is 2, average number of audit log events per

message is 5, number of day's kept for real-time retention is 2. Then the estimates for the file sizes on base of the daily volumes are the following:

Daily Message File Size (GB) = 20 GB

Daily Audit Log File Size (GB) = 50 GB

Retention Message File Size (GB) = 50 GB

Hourly Peak (Memory) Message Size = 10 GB

2.1.5 Existing Messaging Systems

This section provides a general overview of bank messaging systems (BMS) within a financial institution. The functionality pertaining to each process and link is described along with the traffic flow diagrams.

2.1.5.1 Funds Processing

The BMS provides an easily configurable multi-link interface between the FI's backend Funds Transfer System (FTS) and external depository agencies, the New York Federal Reserve Bank and New York Clearing House. The system supports validation, prioritization and routing of inbound and outbound money and non-money FED and CHIPS funds transfer messages.

Facilities are made available for the monitoring and control of messages through varying views and stages. Displays are provided for:

- Balance breakout

- Proofs
- Queue sets
- Backend/front-end line status
- Front-end/external agency line status
- Credit information
- Error descriptions
- Message retrieval

Message controls are provided for:

- Priority release
- Holds
- Release of holds
- Change credit info
- Turn credit check on/off
- Change queue priority
- Set priority accounts
- Resend of messages to backend
- Resend of messages to external agency
- Contingency mode

Inbound messages are validated to ensure they have today's value date, have been sent to the correct Lterm (intended for FI) and are a valid FED

message type/sub-type. The incoming message then gets processed based on whether it's a Payment, Report, a Reject, or a Batch Validation.

- Payments update the FED credit position, have a front-end sequence number pre-pended and are forwarded to the FTS backend system earmarked as either a high or low dollar amount. The pre-pended sequence number is used by to reconcile messages across the two systems at the end of day.
- Reports have a front-end sequence number pre-pended and are forwarded to the FTS backend system earmarked accordingly.
- When the FRB rejects outgoing FI funds transfers the reject message includes a copy of the rejected message along with the rejection reason. The reject message is matched to a saved copy of the original message, a front-end sequence number is pre-pended and the reject message is forwarded to the backend. The reject contains the FED's IMAD, which allows the backend to match the FED reject to the original outgoing message. The front-end application also adjusts the credit position.
- The FRB sends batch validation (Batchval) messages (acknowledgements) for a group (currently 10) of outgoing FI funds transfers. Each acknowledgement in the batch is matched to a saved copy of the original message. Once matched, a front-end sequence number is pre-pended and the Batchval message is forwarded to the backend. The Batchval contains

the FED's IMAD, which allows the backend to match the FED acknowledgement to the original outgoing message.

Outbound messages come across from FTS on either of two lines. One line is for normal day processing and the second line is for messages intended for end of day (EOD) processing. Messages are checked to determine if they are a credit or non-credit check message. The message is then queued depending upon if it's a payment (credit check) or a report/inquiry (non-credit check). Message prioritization is inherent to the queue the message is on. Queue's have default start of day priority settings, but can be changed during the day. Non-credit messages have the highest priority and are sent out immediately. Payment prioritization is based upon: high / low dollar amount, if the account is flagged as a Super Priority, if the account is flagged as a Priority or if the message is a Foreign Transfer.

Before the messages are released to the FRB they are validated to ensure they are a legitimate FED type / sub-type. They then go through a credit check, and a copy of the message is put on a wait acknowledgement queue (for Batchval's), An IMAD is generated for the FRB message and an internally generated front-end acknowledgement (FEACK) is sent to the backend FTS system. The FEACK notifies the backend that the payment has been delivered to the FED and allows the backend to match its original outgoing message to the message at the FED.

The system also provides for automatic recovery of messages lost during line / transmission problems and provides the capability to manually resend messages to the FRB.

2.1.5.2 CHIPS Processing

Inbound messages are validated to ensure they have today's value date, have been sent to the correct Lterm (intended for FI) and are a valid CHIPS message type. The incoming message then gets queued based on whether it's a Payment, Report, Reject or an Acknowledgement.

- Payments are forwarded to the FTS backend system earmarked as either a high or low dollar amount.
- Reports are forwarded to the FTS backend system earmarked accordingly.
- Rejects - When the NYCH rejects outgoing FI funds transfers the reject message includes a copy of the rejected message along with the rejection reason. These messages are matched to a copy of the original message held in the front-end messaging application and forwarded to the backend. The reject contains the CHIPS sequence numbers allowing the backend to match the CHIPS reject back to the original outgoing message.
- Store Acknowledgement - The first level of acknowledgement is at the message delivery notification stage. CHIPS sends a successful store acknowledgement on receipt of each outgoing CHIPS message. The front-end messaging application compares the incoming stored delivery

notification acknowledgement to a copy of the original stored message. Once matched, the store acknowledgement message is forwarded to the backend. The acknowledgement contains the CHIPS sequence numbers. This allows the backend to match the CHIPS acknowledgement back to the original message.

- Resolver Acknowledgement - The second level of acknowledgement is at the message transmittal stage. CHIPS sends a successful funds transmittal acknowledgement on the release of each outgoing CHIPS message to the designated member bank. The front-end messaging application compares the incoming resolver acknowledgement to a copy of the original stored message. Once matched, the resolver acknowledgement message is forwarded to the backend. The acknowledgement contains the CHIPS sequence numbers that allow the backend to match the CHIPS acknowledgement back to the original message.

If CHIPS messages have not received an acknowledgement from the NY Clearing House within a specified amount of time, the message is automatically resent by the front-end as a 'PDM'.

Outbound messages come across from FTS and are queued depending upon if it's a payment, report/inquiry or administrative. Message prioritization is inherent to the queue the message is on. Queues have default priority settings and can be changed for the day by Operations. Non-money messages have the highest priority and are sent out immediately. Payment prioritization is based

upon: high / low dollar amount, if the account is flagged as a Priority or if the message is Administrative.

Before the messages are released to the NYCH they are validated to ensure they are a legitimate CHIPS message type and a copy of the message is put on a wait acknowledgement queue for matching against Store and Resolver acknowledgements.

The system also has the functionality to recover messages lost during line / transmission problems and to manually resend messages to the NYCH.

Credit Checking functionality used before CHIPS Finality is no longer required for CHIPS processing. However, Operations likes to retain this functionality for contingency purposes.

2.1.5.3 Securities Processing.

The BMS provides an easily configurable multi-link interface between the FI's backend Broker Dealer System (BDS) and the various Federal Reserve Bank systems. The system supports validation, prioritization and routing of inbound and outbound money and non-money FED securities messages for the FI and on behalf of surrogate banks.

Facilities are made available for the monitoring and control of messages through varying views and stages. Displays are provided for: proofs, queue sets, backend/front-end line status, front-end/external agency line status, FED report requests, internally routed and message retrieval. Message controls are

provided for: routing repair, FED table maintenance, resend of messages to external agency and contingency mode.

Inbound messages are validated to ensure they have today's value date, have been sent to the correct LTerm (intended for FI or one of the surrogates) and are a valid FED message type/sub-type. The incoming message then gets queued depending upon if it is a Report, Securities, Reject or a Batch Validation message.

- Securities are forwarded to the FTS backend system earmarked as either a high or low dollar amount.
- Reports are forwarded to the FTS backend system earmarked accordingly.
- When the FRB rejects outgoing FI securities the reject message includes a copy of the rejected message along with the rejection reason. These messages are matched to a copy of the original message held in the front-end messaging application. Once matched, the reject messages are forwarded to the backend. The reject contains the FED's IMAD allowing the backend to match the FED reject back to the original outgoing message.
- The FRB sends batch validation (Batchval) messages (acknowledgements) for a group of outgoing securities. These messages are matched to a copy of the original message held in the front-end messaging application. Once matched, the Batchval messages are forwarded to the backend. The Batchval contains the FED's IMAD allowing the backend to match the FED acknowledgement back to the original outgoing message.

Message prioritization is inherent to the queue the message is on. Queues have default priority settings and can be changed during the day. Non-money messages have the highest priority and are sent out immediately. Securities prioritization is based on high / low dollar amount.

Before messages are released to the FRB they are validated to ensure they are a legitimate FED message type / sub-type, and a copy of the message is put on a wait acknowledgement queue. An IMAD for the FRB message is generated, and an internally generated front-end acknowledgement (FEACK) is sent to the backend system. The FEACK notifies the backend that the payment has been delivered to the FED and allows the backend to match its original outgoing message to the message at the FED.

The system also performs automatic recovery of messages lost during line / transmission problems and a manual facility allows an operator to resend messages to the FRB.

2.1.6 SWIFT, TELEX, and FAX

BMS exchanges messages with back-end systems in SWIFT format. Back-end applications that are not equipped to construct or parse SWIFT messages communicate to BMS through MTE (Message Transformation Engine), which transforms messages between SWIFT format and a format produced or expected by the back-end application. Inter-branch traffic is

handled in one of two ways. If the message is destined for a branch that is also connected to BMS, BMS sends the message to IBR (Inter Bank Router), a process which simulates a SWIFT interface by returning an acknowledgment as well as a SWIFT outgoing message (i.e. in the format a message is received from SWIFT). The message is then forwarded to the destination branch. If the message is destined for a branch that is connected to BMS, it is placed on a queue and BMS takes responsibility for delivering the message to the destination branch. Similarly, outgoing messages received or generated by BMS that need to be delivered via TELEX or FAX are sent to the target destination by IC.

On the external side, in addition to SWIFT, TELEX, and FAX, BMS supports direct interfaces ("CPU links") with a number of correspondents. The method of delivery of a message is specified by the back-end system prior to transmission to BMS. Using the CIF record of the target destination, BMS verifies that all identifiers and authentication capabilities are in place as required by the external interface. If all such conditions are not met, BMS may attempt to deliver the message by an alternative method of delivery. Typically, SWIFT is the delivery method of choice, followed by TELEX. Messages to be delivered by FAX must be explicitly requested.

Another aspect of the routing decision is to select a priority. BMS determines the delivery method and priority based on message type,

destination, and other field contents, so that certain business messages will be delivered as quickly as possible.

Outgoing TELEX messages are sent in ISO TELEX format or as free text. Messages that require testing contain test key parameters sent by the back-end in the pass area that are used in the test key calculation prior to transmission.

2.1.7 Trends and Business Drivers

There are two significant reasons why a new messaging infrastructure is needed by FI.

The first and major reason is the age of the existing systems. Most processes that manage financial messaging at FI were built over fifteen years ago using a transactional development platforms which are no longer supported.

The second important reason comes from new and future developments in the SWIFT network. SWIFT's ongoing migration from the X.25 interface to SWIFTNet promises to provide a much more powerful and versatile way for banks to communicate with each other, their customers, and business partners. In addition to the well-established FIN application interface and a revamped file transfer facility, SWIFTNet is providing transport services for CLS (Continuous Linked Settlement), along with capabilities for other closed user groups and web interfaces. It is critical that The FI's messaging and gateway

infrastructure accommodate a flexible means for old and new applications to take full advantage of SWIFTNet connectivity.

2.1.7.1 Message Processing

This section describes the processing steps required as messages are passed from the back-end to an external party, or vice versa. The functional categories (presented here as sub-headings) were chosen to show that a messaging front-end generally behaves the same way regardless of the contents of the message; the same steps are followed even when the processing within those steps are specific to the business event that the message represents.

Messages are passed between back end applications and the communications gateways using one of the following protocols:

- IBM WebSphere MQ, formerly MQSeries, integrates more than 35 platforms, provides the base messaging functions for servers and clients, and assures once only message delivery. MQ (see below) is the messaging standard for The FI.
- SNA LU0 is a protocol that is widely used in mainframe communications.
- SNA LU6.2 is a peer-to-peer protocol that is used to move messages between mainframe CICS applications and the external SWIFT/TELEX gateway.
- NDM (Network Data Mover) is a file-transfer protocol that is widely used at FI to pass files of messages between FI applications. NDM API's,

libraries, and utilities are packaged and sold by Sterling Commerce as Connect Direct.

Application protocol describes the rules and conventions for establishing and maintaining a session for exchanging messages using one of the protocols listed in the previous section. This includes a description of messages that are used to start a session, recover and synchronize, and terminate, as well as any headers or envelopes that contain information related to the session, such as sequence numbers. This discussion of application protocol is only concerned with point-to-point communications between the back-end application and the front-end. End to end protocols (such as the receipt of a notification of delivery by the ultimate receiving application) are excluded here, and are considered to be in the domain of message routing.

In general, back end interfaces impose minimal handshaking, and the major application protocol function is to check sequence numbers to ensure that all messages are received. Moreover, some multi-layered transports (WebSphere MQ, for example) provide a configurable way to adjust the protocol to specify, for example, whether or not the receiving queue manager will return a Confirmation of Arrival (COA) or Confirmation of Delivery (COD) to the sending application.

The following table depicts the required backend / front-end application protocol support:

METHOD OF DELIVERY	INTERNAL COMMUNICATIONS BACK END / FRONTEND APPLICATION PROTOCOL
FEDWIRE FUNDS CHIPS FUNDS	Inbound from Backend No automatic sequence number gap checking is performed. The front-end appends a sequence number to each message passed onto the backend. At the end of day the backend runs a gap report to reconcile. Outbound to Backend No application sequence number checking is performed. Front End Acknowledgement (FEACK), containing the IMAD of the message sent to the FRB, is sent to the back end application
FEDWIRE SECURITIES	Sequence number checking is performed between the backend and front-end
SWIFT/TELEX/FAX	Messages and files received from back end interfaces contain two sequence numbers: the session sequence number and the application sequence number (ASN). Both are checked against the expected sequence number. If a sequence number discrepancy is discovered (i.e. the gap is greater than 25 for messages or 1 for files), the interface is brought down and sequence numbers must be reset and synchronized manually.
SWIFT	BMS verifies a single sequence number contained in the received message, and compares any gap to a configurable gap tolerance. When communicating via WebSphere MQ, BMS relies on the receipt of COA and COD notifications to confirm that a message has reached its final destination.

REQUIREMENTS SUMMARY – APPLICATION PROTOCOL
Backend / Front-end
Support up to two sequence numbers per interface, performing gap checking of each against a configured gap tolerance

2.1.7.2 Message Syntax

As a rule, back end applications determine the delivery criteria for a message, including destination and carrier. Once the method of delivery is determined, these applications create messages in the format required by the external carrier (e.g. FEDWIRE or CHIPS for U.S. domestic payments). While this approach simplifies the work performed by the front-end applications, it imposes a significant burden on back end application developers, who must master the details of the external syntax (particularly SWIFT) and accommodate changes in message standards as they are defined and deployed.

In certain cases, back end applications send messages destined for SWIFT to MTE (Message Transformation Engine) in a proprietary format and MTE transforms the message to SWIFT syntax before passing it along to the gateway (BMS). Similarly, BMS routes a large number of incoming SWIFT messages to MTE for transformation, routing, and delivery to back end applications.

BMS requires the back end application to provide a "pass area" that contains supplementary information about the outgoing message and the target

destination. The fields contained in the pass area are used to route the message and set transmit priority.

Inbound messages are sometimes delivered to back end applications in the format received, in one of a limited number of supported formats (see table below), or they are sent to MTE for transformation to a format suitable for processing by the target back end application. BMS appends a header with additional information.

A major objective of this project is to have no or as little as possible impact on backend applications. The following table shows the message syntax currently used by back-end applications for message exchange for each different method of delivery:

METHOD OF DELIVERY	INTERNAL COMMUNICATIONS BACK END MESSAGE SYNTAX
FEDWIRE FUNDS	FED formatted message, plus header: <u>Inbound Header</u> Messaging application sequence number Backend - In message type indicator Outbound Header 'LLZZ' - LL-Segment length, ZZ - Control bytes Account number Transaction reference number (TRN) Sequence number $ Amount MiscFlag - identify 'CLS' messages (SuperPri) MID-ID - type / sub-type Version number
CHIPS FUNDS	CHIPS formatted message, plus header:

METHOD OF DELIVERY	INTERNAL COMMUNICATIONS BACK END MESSAGE SYNTAX
	Inbound Header CHIPS transaction numbers: PSN, SSN, ISN, OSN, RSN Backend - In message type indicator Outbound Header 'LLZZ' - LL-Segment length, ZZ - Control bytes $ Amount Transaction reference number (TRN) Version number
SWIFT/TELEX/FAX OUTBOUND	In addition to message text, all outbound messages are sent to the gateway with a pass area header that is used to prepare and route the message. The following text formats are currently in use: Inbound PRINT (for mainframe printers) SWIFT II Outbound SWIFT (blocks 1, 2, 4) FI Internal SWIFT (block 4 only) BMS ESA (SWIFT format with Pass Area at the end) Report format (for free format messages)
SWIFT	BMS requires that all outbound messages be in SWIFT II format.
SWIFT (via MTE and BMS)	MTE exchanges messages with back-end systems in a wide range of formats, including EDI, COBOL copybooks. MTE transforms these messages into SWIFT format before forwarding them on to BMS, and transforms inbound SWIFT messages to the format required by the back-end application.

2.1.7.3 Message Evaluation and Disposition

Since messages from back end applications are sent in the format expected by the target method of delivery, text validation is performed by the gateway to assure that the syntax is correct. This syntactical validation ensures that formatting rules for the message type have been followed.

METHOD OF DELIVERY	MESSAGE EVALUATION & DISPOSITION VALIDATION
FEDWIRE FUNDS CHIPS FUNDS FEDWIRE SECURITIES	Cross-checked against rules-based hash tables. These tables have information for every message type (and subtype for FED). Inbound/Outbound The message is checked against the tables to ensure it is a valid message type. Messages that fail edit-check are put in the 'Exceptions' queue. Data content is not validated. Inbound The message goes through consistency checks to ensure it contains the correct: Value Date LTerm Outbound The message is checked against the tables to determine if the format/syntax (e.g. field order, field length) is valid for that message type.
SWIFT / ISO TELEX	Both inbound and outbound messages in SWIFT format are checked to ensure that they conform to SWIFT specifications in the following ways:

METHOD OF DELIVERY	MESSAGE EVALUATION & DISPOSITION VALIDATION
	All mandatory fields are present All fields are in the correct order The data in each field is of the correct data type SWIFT code words applied correctly throughout Currency codes are checked against a table of valid codes An OFAC check is performed for some message types, where the entire text of the message is scanned to determine if one of a list of pre-defined keywords is present. For files received via SWIFT IFT, the contents of the file are checked against the expected syntax (currently either SWIFT message type 102 or a fixed-length format called "TIPA". In addition, when the file contains multiple payments, the sum of the amounts of the individual payments is compared against a payment total contained in the file.
SWIFT (MTE/BMS)	Outbound messages that pass through MTE on their way to BMS are validated to ensure that message data is complete and that fields are of the correct data type. Inbound SWIFT messages that pass through MTE are parsed and validated to ensure that each message fully conforms to the SWIFT specifications.
FREE-FORMAT TELEX AND FAX	No validation is performed, other than to scan the text for OFAC keywords.

REQUIREMENT SUMMARY - MESSAGE VALIDATION
Validate message syntax against the specification or standard to ensure that: All mandatory fields are present Fields are in the correct sequence Field contents are of the correct data type Keywords are used correctly Valid currency codes are used
Perform regulatory checking (OFAC) by scanning the message text to find any of a configured list of keywords. If a keyword is found, the message must be routed to an operator for disposition
Received files that contain multiple payments are to have the sum of the amounts of the individual payments compared against the payment total contained in the file.

Message enrichment is used to get additional data related to a given message. This additional data allows business or routing decisions to be made, or provides data that is necessary for message transformation. Typically enrichment is done by using one or more fields contained in the message as a key to access a data base table to get additional information.

METHOD OF DELIVERY	MESSAGE EVALUATION & DISPOSITION INBOUND ENRICHMENT SOURCES (REFERENCE DATA)
FEDWIRE FUNDS FEDWIRE SECURITIES	Messaging application sequence number Backend - In message type indicator Front-end Acknowledgement (FEACK) Deposit / Securities ($amount – in/hi) Reports Batch Validation
CHIPS FUNDS	Messaging application sequence number Backend - In message type indicator Deposit ($amount – in/hi) Reports
SWIFT/TELEX/FAX	The sender's CIF record is read and used to populate header fields

METHOD OF DELIVERY	MESSAGE EVALUATION & DISPOSITION OUTBOUND ENRICHMENT SOURCES (REFERENCE DATA)
FEDWIRE FUNDS FEDWIRE SECURITIES	IMAD Test or Production system indicator
CHIPS FUNDS	CHIPS transaction numbers: PSN, SSN, ISN, OSN, RSN
SWIFT/TELEX/FAX	Read routing info from CIF using CIF ID or Name Key from Pass Area

REQUIREMENTS SUMMARY - ROUTING AND PRIORITIZATION
Messages are routed based on the field contents of the message and enrichment data (a facility must allow a set of message characteristics (i.e. field names with associated values) to be associated with a destination and priority
Routing must support the ability to route a message to more than one destination, in more than one format
Routing rules are likely to number in the thousands, so the routing algorithm must be fast and scalable to the number of rules
SWIFT, TELEX and FAX
Messages received from back end applications may contain a field that specifies the preferred method of delivery as a "best method" value, in which case the message is routed based on bank preference (SWIFT, then TELEX), and the CIF data for the destination correspondent is checked to ensure that addresses (SWIFT TID, TELEX Answerback, or FAX number) and authentication relationships (authentication keys for SWIFT, a test key agreement for TELEX messages, a digital signature for FAX) are present when the message requires testing/authentication.

OUTBOUND TELEX
Additional routing decisions are made to determine which of four carriers will actually receive the message and take responsibility for its delivery. The business reasons for this are twofold – to get the best rate for individual messages based on the cost of transmission to a particular country, and to generate enough total traffic for each carrier to maintain a business relationship that will keep rates low across the board.
To ensure that the best rate is obtained for a message routed to a particular location, a table is maintained that excludes certain carriers from delivering messages to TELEX numbers with certain country codes, and also indicates if there is a message size restriction for outbound messages. To balance the load across carriers, another table defines the optimum distribution of traffic across the carriers (represented as percentages). The routing determination of outbound TELEX messages adheres to the exclusions in the country code table, and then tries to balance the outbound TELEX traffic to approximate the target percentages.

Incoming messages are generally routed based on fields contained in the message, including message type (or embedded message type for SWIFT type N92, n95 or n96), FI branch code, or any other field/value combination. Incoming payments received from CHIPS or FEDWIRE are funneled directly to a dedicated queue. Incoming logical acknowledgements, which are sent by external destinations to confirm receipt of transmitted messages, may be matched with their corresponding outgoing messages before being routed to a back end application or printer.

METHOD OF DELIVERY	MESSAGE EVALUATION & DISPOSITION INBOUND ROUTING CRITERIA
FEDWIRE FUNDS CHIPS FUNDS	Although current payment systems are not making use of it, automated routing tables can be used to send messages to different destinations based on the contents of the message. This feature should be included in any future front-end system. Incoming messages from the FRB or NYCH are forwarded to the FTS backend system through a single gateway / link on each side. Different message types are placed on separate queues for the purpose of monitoring, reporting and demarcation in the backend application. They are categorized as follows: FED Funds Deposit <= to $1 million Deposit > then $1 million Non-money messages Batch Validations and Rejects CHIPS Funds Deposit low $ amount Deposit hi $ amount Non-money messages Acknowledgements and Rejects
FEDWIRE SECURITIES	Destination routing is determined within the front-end application for FI securities and is predetermined for the surrogate banks. Incoming messages from the FRB come in through specific Lterms, corresponding to eight Fedlink processes. These links correlate one-

METHOD OF DELIVERY	MESSAGE EVALUATION & DISPOSITION INBOUND ROUTING CRITERIA
	for-one to the queues used to move messages between the backend and the front-end. A specified surrogate message goes onto a specific queue. FI messages will be put onto one of two FI queues depending on its associated account.
SWIFT/TELEX/FAX	Messages are routed based on FI branch code, message type, embedded message type, or any other specific field contents specified in the routing table. In addition, messages, which are free format (that would normally be sent to BMS for scissoring and routing), are scanned by a special routine ("power route") that scans the message text for keywords. If a match is found, the message is routed to the associated routing destination.

For the most part, outgoing messages are pre-routed by the back end applications. Outbound messages received by the gateway from the back end applications are pre-formatted and the preferred method of delivery is explicitly indicated either by default (i.e. a dedicated queue or interface that is a pipe for a particular external interface), or by a field included in the header (i.e. the BMS's pass area). Outbound payments are formatted and queued by the back end applications for delivery to either FEDWIRE or CHIPS. For all outbound methods of delivery, messages can be queued to one of several

priority levels, so that critical business messages can precede less important ones. The highest priority is reserved for messages destined for preferred customers, as indicated in priority account tables. This allows FI to provide the highest level of service to those customers.

When the target destination is another FI branch, the external network is bypassed and the message is sent directly to the destination branch. When this happens, Block 2 of the message must be transformed from a SWIFT Input message (i.e. the format of a message to be sent to SWIFT) to a SWIFT Output message (i.e. the format of a message received from SWIFT) before being passed on to the FI branch. In BMS, this transformation is performed in the normal workflow, whereas in BMS the message is "sent" to another application (IBR) that simulates SWIFT by sending back both a logical acknowledgement and the transformed message.

METHOD OF DELIVERY	MESSAGE EVALUATION & DISPOSITION OUTBOUND ROUTING CRITERIA
FEDWIRE FUNDS CHIPS FUNDS	Although current payment systems are not making use of it, automated routing tables can be used to send messages to different destinations based on the contents of the message. This feature should be included in any future front-end system.

METHOD OF DELIVERY	MESSAGE EVALUATION & DISPOSITION OUTBOUND ROUTING CRITERIA
	Outgoing messages from the FTS backend are forwarded to the FRB or NYCH through a single gateway / link on each side. Messages are placed on different queues based on priority for the purpose of monitoring and control, liquidity management, reporting and release prioritization to the outside agencies. Priorities are assigned to queues, so that a message's priority is inherent to the queue it is on. Operator should have the ability to change queue priorities during the day. Start of day defaults are table driven. Fed messages are routed to various queue based upon: Sending line message came across on Credit or non-credit check message Message types Priority $ Amount CHIPS messages are routed to various queue based upon: Sending line message came across on Money or non-money message Message types Priority $ Amount

METHOD OF DELIVERY	MESSAGE EVALUATION & DISPOSITION OUTBOUND ROUTING CRITERIA
FEDWIRE SECURITIES	Destination routing is determined within the front-end application for FI securities and is predetermined for the surrogate banks. Outgoing messages from the backend come in through eight separate queues. A specific queue is designated for each of the six surrogates and two for FI. The surrogate queues correspond one-for-one to six Fedlink processes. Messages from either of the two FI queues will be routed to one of two FI Fedlink processes depending upon its associated account. Outbound messages that can be internally routed have a "dummy" receipt message generated, which include front-end generated IMAD and OMAD numbers. Acknowledgements are also created. In addition, messages are segregated on to queues for the purpose of monitoring and control, reporting and finally release prioritization to the regional FRB's. Securities messages are routed to various queues based upon: link message came across on Securities or non-securities message Message types $ Amount Priorities are assigned at the queue level. Therefore, message prioritization is inherent to the queue

METHOD OF DELIVERY	MESSAGE EVALUATION & DISPOSITION OUTBOUND ROUTING CRITERIA
	it is on. Operator should have the ability to change queue priorities during the day. Start of day defaults are table driven.
SWIFT/TELEX/FAX	Messages received from back end applications contain a field that specifies the preferred method of delivery. This field can also contain a "best method" value, in which case the message is routed based on bank preference (SWIFT, then TELEX), and the CIF data for the destination correspondent is checked to ensure that addresses (SWIFT TID, TELEX Answerback, or FAX number) and authentication relationships (authentication keys for SWIFT, a test key agreement for TELEX messages, a digital signature for FAX) are present when the message requires testing/authentication.
TELEX	Additional routing decisions are made to determine which of four carriers will actually receive the message and take responsibility for its delivery. The business reasons for this are twofold – to get the best rate for individual messages based on the cost of transmission to a particular country, and to generate enough total traffic for each carrier to maintain a business relationship that will keep rates low across the board. To ensure that the best rate is obtained for a message routed to a particular location, a table is

METHOD OF DELIVERY	MESSAGE EVALUATION & DISPOSITION OUTBOUND ROUTING CRITERIA
	maintained that excludes certain carriers from delivering messages to TELEX numbers with certain country codes, and also indicates if there is a message size restriction for outbound messages. To balance the load across carriers, another table defines the optimum distribution of traffic across the carriers (represented as percentages). The routing determination of outbound TELEX messages adheres to the exclusions in the country code table, and then tries to balance the outbound TELEX traffic to approximate the target percentages.

Requirements summary for routing and prioritization is the same as above.

The general philosophy of the existing message infrastructure assumes that back end applications can send and receive messages in the format specified by the external interface. FTS sends and receives messages in FEDWIRE or CHIPS format, depending on the source or target of the payment.

METHOD OF DELIVERY	MESSAGE EVALUATION & DISPOSITION INBOUND TRANSFORMATIONS
FEDWIRE FUNDS CHIPS FUNDS	Received messages from either the FRB or NYCH do not require any transformation. The funds transfer front-end and backend systems utilize the FED and CHIPS formats. Message header is added to reflect the appropriate message type and sequence number. Message converted from ASCII to EBCIDIC.
SWIFT/ISO TELEX (BMS)	All received messages are transformed into a canonical format to maximize commonality for key processing functions, such as persisting the message and routing. After the routing destination is determined, the message is transformed again to the syntax required by the receiving destination.
SWIFT (BMS)	A number of inbound SWIFT messages are routed through the MTE system, where messages are transformed from SWIFT format to the back end's format (COBOL copybook, e.g.).
FREE-FORMAT TELEX	Received messages that are free-format cannot be parsed or routed. They are routed directly to BMS workstations for manual routing and scissoring.

Note that the transformations listed above are based on formats received on one end and delivered on the other. In practice, each source format may be transformed to an internal, canonical format (or multiple canonical formats, if, for example, an internal format specific to payments is used. The use of a

canonical format simplifies the implementation of enrichment, routing rules, and any other semantic logic that may be required prior to delivering the message to its destination. If an internal format is used, then the list of transformations (as listed above) would change, with one transformation from each source format to canonical format, and one transformation from canonical format to each target format.

The primary external interfaces supported by the FI front ends are SWIFT, CHIPS, FEDWIRE (Funds and Securities), TELEX, and FAX (outbound only). In addition, point-to-point interfaces connect FI directly to a number of customers and correspondents either to save cost, provide a higher service level, or to provide outsourcing functionality.

It is a mandatory requirement that vendor solution(s) encompass a current standing certification with the external agencies. This inherently assures the compliance to external protocol, security and syntax.

METHOD OF DELIVERY	EXTERNAL COMMUNICATION COMMUNICATIONS PROTOCOL
FEDWIRE FUNDS FEDWIRE SECURITIES	FLASH utilizing SNA LU0 (CERTIFIED)
CHIPS FUNDS	X.25 protocol (CERTIFIED)
SWIFT	Currently SWIFT uses a layered protocol built on a X.25 protocol which is being replaced by IP. (CERTIFIED)
TELEX/FAX	X.25 to TELEX/FAX carriers today, though an IP interface would be preferable in future systems
CPU LINKS (Point-to-point)	WebSphere MQ, SNA LU6.2

2.1.7.4 Security

METHOD OF DELIVERY	EXTERNAL COMMUNICATION SECURITY AND AUTHENTICATION
FEDWIRE FUNDS FEDWIRE SECURITIES	No authentication keys are used. Signon required per session with password. All traffic is encrypted.
CHIPS FUNDS	Authentication keys are used. All traffic is encrypted.
SWIFT	At the session level, login messages must contain tightly controlled login keys. Message traffic on SWIFT is always encrypted. SWIFT also requires correspondents to authenticate many messages, particularly when money will be moved based on message contents. Authentication is done using a special algorithm that calculates a check sum based on characters in the message and the values of a key string that is known only by the two banks. These keys are exchanged and maintained

METHOD OF DELIVERY	EXTERNAL COMMUNICATION SECURITY AND AUTHENTICATION
	using SWIFT's BKE facility. SWIFTNet institutes a public key infrastructure (PKI) that, according to SWIFT, "offers provable authenticity of messages, non-repudiation of the origin of messages, trusted time stamps, and individual encryption of messages". Though the BKE will still be used by FIN while institutions are migrating to SWIFTNet, it is expected the BKE will be supplanted by PKI sometime after the entire community is connected to SWIFTNet.
TELEX	TELEX messages may be tested or untested, depending on the business context of the message. A tested message contains a test key that is calculated using an algorithm agreed by the sender and receiver, based on the contents of specific fields contained in the message. Managing and processing test keys is a complex process, and a complete description of functionality is beyond the scope of this document. Test key calculation must support test key calculation as well as the persistent maintenance of test key history for each correspondent, and must be invokable both from the real-time TELEX gateway and manual test key functions.
FAX	Outgoing FAX messages contain a signature key that is inserted into the message prior to transmission.

2.1.8 Special Cases

2.1.8.1 File Transfers

SWIFT provides a facility to transfer files across their network called IFT (Interbank File Transfer). At present, IFT is used at FI to receive files from a small number of customers. These files contain multiple payment orders that are forwarded to back-end systems for processing as checks or wires. Within the front end, processing of received files consists of the following steps:

- Validate the received file to ensure that component messages conform to the expected syntax (SWIFT MT102 and fixed-length "TIPA" format are the two formats currently in use), including field sequence and presence of manual fields.
- Calculate the sum of all component payment messages and verify that the total provided in the message is accurate
- Construct and append a header, including a sequence number that is used to ensure that all messages are processed.
- Forward the file to the back-end system, based on a configuration table that matches sender with target system, directory, and file name.

This last step, the transfer of the file to the back end, requires some coordination between the sender and receiver, to ensure that the back-end successfully processes one file before receiving another. This is provided for in the existing system by requiring the system receiving a file to send an "ACK"

file back to the sender when processing of a file is complete, which serves as a trigger for the next file to be sent.

SWIFT has stated that IFT will not be supported beyond October 2003; FileAct is slated to be its successor.

As volumes increase, the flexible ability to invoke efficient and reliable file transfer facilities rather than send individual messages will allow FI to maximize message throughput economically. The front end should have the ability to collect messages as they are received, format a file, and transmit the file to the receiving destination. Because this facility can only be used for messages whose latency tolerance is very high (i.e. low priority messages), this aggregation capability must be highly configurable, taking into account the following factors:

- the ability of the receiving party to receive a file
- the message type(s) to be aggregated
- the amount of time between file transfers (or maximum record count, if applicable)

Conversely, if the receiving destination is not capable of receiving and processing a file, the front end must be able to break a received file into component messages, and validate, route, and transform each one individually.

REQUIREMENTS SUMMARY – FILE TRANSFERS
File Transfers
Send and receive files to/from selected back-end systems
Send and receive files to/from selected external systems
Optionally separate received files into individual messages before forwarding
Aggregate received messages into files, based on destination, message type, time limit, and message limit

The FI supports direct interfaces with a number of customers and correspondents as an alternative to SWIFT or TELEX. The ability to channel traffic directly allows FI to provide a higher level of service, but requires special processing to handle proprietary protocols and message formats. In addition, FI needs a way to ensure that business messages from CPU link correspondents (among others) are processed at a higher priority than "general" message traffic.

The FI provides outsourcing services for back office letter of credit processing. To support this in the back office, FI maintains a separate office, systems, and staff that are dedicated to supporting outsourcing business. All messages supporting this service must be traceable to show messages awaiting processing, messages awaiting transmission, and messages awaiting acknowledgment. In addition, all messages relating to outsourcing business must be carried on queues that are dedicated to outsourcing traffic, and must ultimately be sent to applications or devices that are dedicated to outsourcing.

The FI processes Securities on behalf of other Banks. These banks are referred to as surrogates. Surrogates are segregated from each other, as well as the FI.

Surrogate messages can be viewed through the Access (iAccess) screens. Each surrogate has its own set of screens to monitor queues, proof inbound and outbound messages; request FED reports and has a facility to resend messages to the FED. The screen displays show message totals and aggregate amounts by message type, FED line traffic and queue names.

The FI does not provide end of day proof with the FED for the surrogates. That is a surrogate responsibility.

Determines whether the debit party account of a payment has sufficient funds to complete the transaction. Also, after a credit transaction takes place, funds credit checking sees if any payments are now eligible for release due to the increased balance of the customer's account.

The FI must maintain a predetermined credit position with the Federal Reserve Bank, based on a moving two week self assessment average. The Funds Transfer messaging system manages the release of funds to the FRB based upon system wide credit limits and deposits received by the FI.

Every payment message must go through credit check before being released. If there are not enough funds remaining to release the payment, the

payment remains on hold. The payment will beis released automatically when funds to cover the payment are received.

FED Credit Remaining Position = Credit Line + Miscellaneous Balance + Total Messages In – Total Messages Out

- Credit Line is a start of day default, which can be changed by Operations.
- Miscellaneous Balance is the opening position with the FED, plus offline credits. Entered daily by Operations.

There is Super Priority limit maintained in the system for Super Priority customers. The spread between the super priority limit and the credit remaining balance is used as additional credit for the release of payments for these super priority customers.

There is a System Wide Credit Limit maintained in the system, which is used as the upper boundary limit an operator can set the credit limit to. If there were a need to set the limits past the system wide limit, it would require two operator changes. This is a safety net against operator input error.

2.1.8.2 SWIFTNet Gateway

SWIFTNet is SWIFT's advanced next generation IP-based messaging infrastructure. It supports a suite of core services addressing the key requirements for interactive messaging and file transfer. All users of the new Secure IP Network (SIPN) must access it through SWIFT's mandated

SWIFTNet Link (SNL) product, which provides the basic set of network connection services.

Two modes of communication are supported over SWIFTNet. Request/reply messaging, including the FIN application, is supported by InterAct, and file transfers are supported by FileAct. Access to the network should be architected to support numerous applications and application protocols (such as FIN and CLS) to share a single gateway to SWIFTNet.

2.1.8.2.1 InterAct

SWIFTNet InterAct offers secure, real-time interactive communication between two customers within a closed user group. It allows both end users and applications to send messages (requests) to another application and receive an immediate response.

The interactive exchange of messages over InterAct can be performed synchronously or asynchronously, by means of SNL SML-based application programming interface (API) functions, thus accessing the value-added features of InterAct, including:

- Message encryption
- End-to-end data authentication (signing)
- Support for non-repudiation

2.1.8.2.2 FileAct

SWIFTNet FileAct supports the exchange of files between two customers within a closed user group. Transferred files may be in free format or structured in either proprietary-defined or SWIFT-defined formats.

FileAct features include:

- Delivery notification
- Push and pull modes
- File delivery monitoring
- File authentication
- Concurrent file transfers

2.1.8.2.3 CLS

Continuous Linked Settlement (CLS) provides a centralized means for matching and settling foreign exchange trades. At present, CLS is live and in production at FI, and is handling a limited number of trades in line with the global implementation which is slowly adding volume and functionality.

The FI is currently using Fundtech's PAYplus CLS product, which supports and enforces the rules and regulations of the CLS Bank, and facilitates foreign exchange transaction processing. In addition, PAYplus provides CLS liquidity forecasting and monitors risk exposure across the entire FX trading chain.

Messages are sent to the CLS Bank over SWIFTNet, and the current implementation has a "hard-wired" interface to a SWIFTNet gateway that is dedicated to CLS messaging.

In the long term, all applications requiring access to SWIFT should be able to use a single SWIFTNet gateway.

2.1.8.2.4 XML

The financial industry in general, and SWIFT in particular, are beginning to evaluate the potential for using XML as a standard syntax for financial messaging. Messages sent via SWIFTNet contain an XML envelope, but for FIN, the payload is still in SWIFT II format. SWIFT and FIX are in the process of studying the feasibility of merging their standards for securities-related messages, and the resulting standard is expected to use XML.

Despite these initiatives, it appears that the adoption of XML for financial messaging is still a long way off, largely because of the broad acceptance of SWIFT II syntax throughout the financial community for both external and internal communication.

2.1.8.2.5 Closed User Groups

Within the ever-expanding scope of the SWIFT network, there is now a means for SWIFT-member financial institutions to exchange messages with any correspondent (i.e. not necessarily a SWIFT member) over SWIFTNet. Closed user groups can be established for any business purpose, and messages that are exchanged can be in any format agreed upon within the group.

Currently, CLS is using SWIFTNet, and participating members are members of the corresponding closed user group.

The business potential for creating and exploiting closed user groups appears to be boundless. A single SWIFTNet gateway should be able to support FIN, CLS and an extensible number of additional application interfaces that will exchange messages over SWIFTNet using a private network built using SWIFT's closed user group capabilities.

2.1.9 Manual Message Processing

2.1.9.1 Routing and Repair

Free-format incoming SWIFT and TELEX messages that cannot be routed automatically are queued to a BMS workstation for scissoring and routing. Operators at the BMS workstation can scissor a message (i.e. break the message into multiple discreet messages) and route each segment to one or more different printers. If the message does not require scissoring, the message can be routed in its entirety to one or more printers. To ensure that the message is not processed twice when more than one printer destination is indicated, only one printer will receive a given message or message segment that is marked "active". All other printer destinations for that message or message segment receive a copy of the message marked "informational".

BMS also has the ability to link multiple versions of a single message, to allow for language translation and de-garbling of illegible messages, while preserving the original attributes of the message.

Outgoing FEDWIRE Securities messages that fail routing check are queued in the front-end for operator intervention. Once repaired the message continues processing.

REQUIREMENTS SUMMARY – MANUAL ROUTING, SCISSORING, AND REPAIR
Routing
Route to one or more internal destinations (SWIFT, TELEX, FAX)
If multiple destinations, distinguish between primary route and copies (SWIFT, TELEX, FAX)
Route inbound paymentsFED Securities transactions to appropriate back-end
Route outbound paymentsFED Securities transactions to appropriate FLASH session
Scissoring
Allow operator to scissor a message into several separate messages
Require that the whole message be scissored, and each segment be routed
Do not allow overlap
Allow full routing capability for each segment
Provide parent/child linkage for query and audit trail
Repair
Allow translation of message text (language, de-garbling), preserving before and after text images

2.1.9.2 Queries

All existing message systems provide inquiry capabilities for messages sent or received, subject to availability of the message (see Section 2.1.11.3 - Retention).

METHOD OF DELIVERY	MANUAL RETRIEVAL CRITERIA
FEDWIRE FUNDS CHIPS FUNDS FEDWIRE SECURITIES	Message Retrievals Search criteria should be key oriented and include: Message ID Range Queue Thread External Agency Keys (e.g. ISN, OSN, SSN, TRN, ASN, IMAD, OMAD) Input Reference Input Line Input Sequence Range Output Reference Output Line Output Sequence Range Time Range $ Amount Range
SWIFT/TELEX	Search criteria include: Correspondent's SWIFT address Correspondent's Answerback Correspondent's Namekey Test Number ASN ISN or OSN Transaction Reference (TRN – field 20) Message Type Message ID (unique message identifier)
SWIFT	BMS allows message queries by: Date/time Message Type Transaction Reference Number (Field 20) Source or target back-end application Direction (incoming or outgoing) ACK Status Delivery Status

Historical queries are supported by a mainframe archival application and database that receives a daily file of the day's message traffic.

REQUIREMENTS SUMMARY - QUERIES
Selection Criteria
Correspondent's SWIFT address
Correspondent's Answerback
Correspondent's Namekey
Correspondent's Account number
Correspondent's Fed ABA
Correspondent's CHIPS ID
Test Number
ASN
Back-end system name (source or destination)
Back-end sequence number
External interface sequence number
Transaction Reference
Message Type
Message ID (unique message identifier)
Time range
Direction (incoming/outgoing)
ACK status
Delivery Status
Currency
Session identifier (FI SWIFT ID, FRB ID, TELEX Carrier)
External reference (ORN, IRN, OMAD, IMAD, etc.)
Amount range
Query Features
Access to audit trail
Current location or final disposition
Parent/child linkage (for scissored messages)
Access to both files and messages
Formatted Fields
Correspondent
External Session, sequence number, and time sent/received
Back-end application, sequence number, and time sent/received
Currency and amount
Transaction Reference

2.1.9.3 Queue Displays

Queues are used extensively in the existing ALPHA messaging systems and in BMS to manage workflow as a message passes from one process to another, or from a process to a user function such as BMS. Individual queues may be selected and the messages on a queue may be displayed. Options exist to see the entire queue or a specified number of messages from the front of the queue.

REQUIREMENTS SUMMARY – QUEUE DISPLAYS
All queues used between applications and those used internally by applications should be visible
Display should show number of messages on queue
Queues containing domestic payments should have total amount on queue displayed
Queue display should automatically refresh, based on a configurable number of seconds

2.1.9.4 Message Entry

Message entry of all SWIFT message types is currently provided through a customized data entry facility built using BMS EMS. Once captured and approved, these messages may be sent via SWIFT, TELEX, or FAX. Among the special features provided in the customized BMS DEV are:

- Creation and use of templates to facilitate the entry of common business messages
- Integration with the FI CIF to support ID translation and validation

- Automatic generation of a unique transaction reference number (TRN) to be used in SWIFT field 20.
- Ability to enter test key parameters, including the ability to invoke a third party for testing purposes.
- Ability to enter information required for FAX transmission
- Blind key verification of value date, currency and amount for tested/authorized messages
- A customized version used in New York provides the ability to enter additional fields that are inserted into the pass area header for messages that will be sent to SWIFT via BMS.

No such facility for Fed funds, Chips funds and Fed Securities exists within the messaging applications.

REQUIREMENTS SUMMARY – MESSAGE ENTRY
Message Entry (SWIFT, TELEX, FAX only)
Provide for templates – define, store, use
Automatic unique TRN generation
Ability to add test key parameters
Ability to enter 3rd Party testing correspondent and automatically send message
Ability to customize to add header/envelope fields and format header/envelope on release
Provide ID translation for correspondent (configurable, based on CIF keys)
Validate all SWIFT fields, including BIC lookup in all applicable fields
Provide flexible requirements for up to two approvers
Require blind key verification of value date, currency and amount for tested/authorized messages

2.1.9.5 Requeuing

Once incoming messages have been delivered to their final destination, the message becomes inactive (i.e. not on an active queue). In the event of a misroute, the message may be manually reactivated by requeuing it to BMS. An operator may then manually route the message to a new destination printer. A history of the message is retained, so that printed output shows all processing and routing steps from its original entry into the system up to its current location or final destination.

REQUIREMENTS SUMMARY - REQUEUING
Requeuing
Applies only to inactive messages (i.e. messages that have already been delivered)
Select message by any query criteria (see section 0 above)
Selected message may be sent to a printer only
Requeuing action must be recorded in the message audit trail

2.1.9.6 Retransmission

The front-end messaging systems should have a menu driven facility to resend messages to either the backend or to outside agencies. Messages are selected with the following criteria:

- Fed Funds and Securities resend to backend – by ISN (range allowed) for a Data message (Money or Non-money) and by TRN for a FEACK.
- Fed Funds and Securities resend to FRB – by IMAD (range allowed) or TRN

- Chips Funds resend to backend – by OSN
- Chips Funds resend to NYCH – by TRN
- SWIFT by ISN
- TELEX and FAX by outgoing sequence number (NOTE: FAX resends should allow specification of different FAX number and timeout value)
-

REQUIREMENTS SUMMARY - RETRANSMISSION
Retransmission
Select message(s) by sequence range or any query criteria (see section 0 above)
Applies to all internal and external destinations
All retransmissions must incorporate a possible duplicate indicator, according to the interface specification
Retransmit action must be recorded in the message audit trail

2.1.9.7 Manual Message Testing

TELEX messages that fail automatic testing must be tested manually. The message is first queued to the BMS function, where test key parameters can be extracted from the message and submitted for test calculation. If the test succeeds, the message may be routed normally. Otherwise, the message is queued for manual testing.

There are three categories of test keys: FI test key (sometimes referred to as the Irving test key), third party testing, and private. FI testing consists of

extracting test key parameters and submitting them for calculation and insertion in the "bingo" card, which contains a history of test key results for each FI correspondent. Third party testing is a fee-based service, which is requested by correspondents who have not exchanged test keys with a correspondent and are asking FI to act as an intermediary. The procedure is much the same as the FI manual test, but a message is generated to the sender with information about the results of the test. The final category, private must be calculated manually and the result is compared to the test result on the inbound message before the result is stored on the "bingo" card.

Regardless of the outcome of the test key calculation and comparison, the message is routed to the predetermined destination printer, with an indication of whether the test has passed or failed.

Additional features of the FI service testing include:

- Altering the gap tolerance as necessary.
- Pre-allocation of a series of sequence numbers to allow later insertion of third-party test results.
- Summary report showing daily user activity – one by function, another showing activity hour by hour.
- Ability to correct a failed test when a revised message is received.
- Ability to assign testing of one correspondent's messages to another (usually affiliated) correspondent's test history.

REQUIREMENTS SUMMARY – MANUAL MESSAGE TESTING
Manual Message Testing
Provide ability to enter test key parameters
Interface with automated testing engine to get/test result (see Section 0)
Allow operator to change gap tolerance for correspondent
Reserve sequence numbers for later update
Allow updates to test slots to amend results
All actions on messages should be appended to message audit trail

2.1.9.8 BankFAX

Messages that cannot be delivered by the FAX carrier are rejected and sent to a queue. A user may access messages from that queue, and resubmit them for FAX transmission. In addition, user may change the FAX number and timeout value prior to release.

2.1.10 Monitoring and Control

This section discusses the tools and facilities that allow an operator to see what is happening in the system, and to control devices and message flows as necessary to work around problems and keep the components of the system running smoothly. Because there may be a large number of disparate components within the messaging front end, it is critical that all component applications support these features with an API, so that a single console or dashboard can be built to manage all components of the system from a single workstation.

2.1.10.1 Logging

Logging is used to provide a chronological record of significant events of a system, typically used in troubleshooting to understand the context of a system or program malfunction. To create this context, log entries should be made for normal as well as abnormal events. The normal events should generally identify state changes in a system resource (e.g. a line, a session, a process, or a device) or a manual action that affects the system environment or a particular message.

Log entries should contain the following fields, wherever applicable:

- Timestamp
- Message identifier

- Operator identifier
- Line/Session/Device identifier
- Severity level

The following is a summary of events that must be logged, grouped by functional category:

- Communications Interfaces and Devices
- Printers
- Printer down/up/unavailable
- Printer redirection
- Communications Interfaces
  - Session up/down, login/logout
  - Sequence number error or reset
  - Resends
  - Retrievals
  - Negative acknowledgment of transmitted message
  - Authentication/Testkey failure
  - Possible duplicate
  - Lterm reconfiguration
  - Alternate routing change
- System events
- Start of day

- End of day
- Program failure
- System Failure / Failover
- Message handling
- Routing failure
- Manual reroute
- Payments
- Priority change
- Credit check on or off
- Credit limit change
- Account priority update
- Securities
- Updates to table associating accounts with FRB branches and back-end systems
- Errors
- Data base
- Queues
- Parsing/Transformation
- Logic and program failure
- Reference Data
- CIF update

- Routing table maintenance
- Other table updates, inserts, deletes

2.1.10.2 Audit Trail

The following events should be recorded and associated with each message to provide an audit trail of automated and manual activity:

- Initial receipt of message
- Message text transformation
- Ultimate delivery(ies) of message
- Any workstation action on message
- Test/Authentication result
- Credit check result
- Routing decision result

Each audit entry should contain the following information (as applicable to message state):

- Time stamp
- Elapsed time since the message was first received (to measure latency)
- Line ID, Session ID, Operator ID, as appropriate
- Message text if a transformation has occurred

2.1.10.3 Tracing and Diagnostics

The following capabilities should be available to assist in problem determination:

- Ability to turn tracing on or off
- Trace transitions between workflow nodes (or entry/exit of processes or tasks)
- Trace all I/O actions and return codes
- Trace routing decisions and other critical decision points

2.1.10.4 Exception Handling

In general, if a message cannot be processed due to an exception discovered during any processing step, the message is routed to an Exception queue or a printer for manual action. Future systems should provide a means to send notifications when exceptions occur, such as a page or email. Individual components should broadcast notifications to a central process using a standard protocol such as SNMP. The central process should in turn generate notifications.

REQUIREMENTS SUMMARY – EXCEPTION HANDLING
All exceptions should be logged at the highest priority
Report any error related to access to data base, queue, line/session, or printer
Any error in processing a message, including parsing and routing, should result in the message being routed to an exception queue.

2.1.10.5 Intervention Controls

Wherever possible, new systems should be able to run in a parallel, load-balancing configuration, where the primary and backup machines share the message load when all systems are operational, but one will pick up the full load if the other fails. In addition, the following features and capabilities should be available to allow operators to work around problems and malfunctions:

- Operators may redirect output from one printer to another.
- Sequence numbers may be reset for all internal and external interfaces
- Messages may be manually placed on a particular queue. Operationally, this generally is used to place a message on the BMS queue for manual routing. In an emergency situation, however, an authorized operator or technician may place the message on any queue.
- Allow redirection of traffic to backup facilities for external communication

2.1.10.6 Monitoring – Performance

No performance monitors are in use on the ALPHA systems other than VMS utilities that track CPU utilization, disk space, memory, and other system resources. . Nonetheless, future systems should be instrumented to measure throughput for all message-handling processes as well as for each communications interface. BMS, which runs under CICS, can be monitored using a variety of tools that CICS provides. The "Current State Report" in

BMS provides a valuable summary of message traffic over an interval of time, but future systems should be instrumented to measure throughput for all message-handling processes as well as for each communications interface. Such a facility should allow an operator to filter on specific interfaces, time intervals, etc. to determine if the selected components of the system are performing satisfactorily, and to produce statistics based on message type, correspondent, etc. In addition, instrumentation should be configurable to provide latency measurements for messages between any two nodes in the system, whether they are adjacent, or whether they are at the extreme ends of the message flow.

REQUIREMENTS SUMMARY – PERFORMANCE MONITORING
All processes in the automated workflow should report at configurable intervals the number of messages processed.
Audit trail should capture time stamps to allow latency measurements

2.1.10.7 Monitoring – Health

In the existing gateways, system health is monitored by facility, which displays internal queues and lines/links that reflect all critical processing states in the system workflow. The queue screen display shows the queue names, number of messages on each queue, and, where applicable, the aggregate

amount of the payments on the queue. When the number of messages on a queue exceeds a configured threshold amount, the operator initiates an investigation of the processes involved to see if a system or program fault has occurred. The line screen display shows the line name, Lterm name, number of messages per Lterm, status (Up/Down) and where applicable the aggregate amount per Lterm. If a line is down the control room operator calls for support.

2.1.10.8 Monitoring and Control – Lines, Sessions, and Devices

REQUIREMENTS SUMMARY – MONITORING AND CONTROL
Lines and Sessions
Display the current state (up/down) of all internal and external interfaces
Display sequence numbers and total messages for the current day for each interface
Allow operator to start and stop any interface
Allow operator to log into any session that requires login
Allow operator to see how many messages were processed in the last n minutes by each interface, where n is a configurable period of time
Printers
Display the current state (up/down) of all printers
Display total messages printed in the current day for each printer
Allow operator to stop and start a printer
Allow operator to redirect messages from one printer to another

REQUIREMENTS SUMMARY – PAYMENTS MONITORING
General
Display pending payments based on a configurable amount ranges
Display current balance across all payment interfaces, listing limits, totals received and sent, pending payments, possible duplicates and unresolved (for CHIPS messages only).
Display items pending (including amount) in the following categories: Awaiting transmission to external agency by priority Payments on hold Pending resent payments Rejected payments awaiting action Awaiting transmission to the back end
Display foreign payments and reversals separately from payments when displaying totals
Show combined message totals and aggregate amounts by message type for FI
Show combined message totals and aggregate amounts by message type for each surrogate

2.1.10.8.1 Control

1. Message Release – allows priority release of messages. Selection by dollar amount, time frame or TRN.
2. Hold Messages – facility to put messages on hold based upon dollar amount, time frame or TRN.
3. Release previously held messages – based upon dollar amount, time frame, TRN or ALL held messages.
4. Turn credit check on or off – always left on for FED funds. Desired functionality for CHIPS, although not required since CHIPS Finality.
5. Change credit info – provides update facility for Operations Control Center:

- Miscellaneous Balance – Fed opening balance + offline credits
- Credit Line – 2 week self assessment average with FED
- Super Priority Limit – higher credit limit for SuperPri accounts
- System Wide Credit Limit - this is an internal safety net credit limit to guard against input error

6. Change workflow priority – allows to dynamically manage message flow by determining the order of next processed.
7. Priority Accounts – allows the ability to identify accounts as a 'priority' or 'super priority' account.
8. Gap Report – produces a sequence gap report for reconciliation.
9. Resend Messages – provides facility to resend messages to the backend or external agencies by agency key or TRN. Messages are flagged as PDM.
10. FLASH Report Generator – allows the operator to request reports from the FRB for:

- Retrievals
- Batch Acknowledgements
- Lterm Grand Totals
- Detail Summary of Transfer Messages
- Securities Cash Summary Request
- Cusip Balance by Sub Account
- Error Code Description

- Repo Balance by Cusip – new requirement

11. RAF Utility (Securities only) – provides statistics of total messages scanned, bypassed, routed to a backend/FRB/all backends or moved to scrap/RAF exception. It also provides the ability to repair routing exceptions.
12. FED Table Maintenance (Securities only) – this table synchronizes to the account residency of GS1 and GS2. The table is used to determine routing of inbound and outbound FI securities to the appropriate backend or FRB Lterm respectively. Actions provided are: add, delete, update and show information in the routing table.
13. Contingency Mode – allows FI to disconnect by Lterm and reconnect to a contingency site.

2.1.10.9 Monitoring – Trends and Patterns

Existing messaging systems at FI provide minimal facilities for viewing and analyzing traffic patterns. Volume reports are available on all systems, broken down by message type, but a more robust traffic analysis would benefit systems analysts in capacity planning, load balancing, and configuration. Further breakdowns by sending/receiving back end application or correspondent could be valuable to business analysts. In addition, if daily traffic is plotted over time, messaging patterns could be derived and checked against daily trends to show when abnormal traffic patterns are occurring.

2.1.11 Daily Checkpoints

2.1.11.1 Start of Day and End of Day

METHOD OF DELIVERY	START OF DAY AND END OF DAY
FEDWIRE FUNDS CHIPS FUNDS FEDWIRE SECURITIES	Processing Timeline 21:30 FTS Start of Day Run 00:02 CHIPS/FED Funds Link Start 00:30 CHIPS Prefunded 00:35 FTS FED/CHIPS Mature 06:30 BDS Login 07:00 CHIPS Member Banks Open 08:30 FRB Banks Open FED Credit Line Set 10:00 FTS CHIPS Mature 15:00 Securities Wire Closes / Settles Free of Payment Period Starts 15:30 Securities Reversals End FED Securities Closes 16:30 CHIPS Send Final Cut-off Notification 17:00 CHIPS Closes 17:30 CHIPS Settles 17:50 FED Funds 2nd Verification Limit Raised 18:00 FED Funds 3rd Party Closes 18:30 FED Funds Closed 19:00 Free of Payment Period End FED Securities Send EOD Reports 20:00 FTS End of Day / Batch
SWIFT/TELEX	The current SWIFT/TELEX systems run around the clock, with only minimal down time to perform housekeeping tasks and reset the run date. Ideally, the system that supports SWIFT and TELEX traffic must be able to run 24 hours no disruption to service for end of day and start of day.

2.1.11.2 Batch Jobs and Reporting

2.1.11.2.1 Scheduler

A number of batch jobs must be run during the course of the business day. Jobs to run reports, backups, purges and the like cannot be run manually on a daily basis – the risk of error is too great.

A robust scheduler utility is required to maintain operational consistency and reliability. The scheduler allows an operator to define what jobs need to be run, what dependencies exist between jobs or between a job and an external event, plus it provides a facility to change the schedule as needed.

2.1.11.2.2 Microfilm

Files are generated daily to store messages on optical disk for long-term storage. The following separate files are generated:

- Incoming SWIFT, TELEX, and FAX
- Outgoing SWIFT, TELEX, and FAX
- BMS routing activity
- Outsourcing traffic
- Weekly traffic summary with a breakdown of messages sent and received by category and message type
- FED, CHIPS and Securities.

2.1.11.2.3 Charging

Messages sent via TELEX or FAX may be charged to either the sending department or to the receiving customer. This determination is made by the originating application, and is passed to the front end in the pass area.

After the message is transmitted, and the carrier has successfully delivered it to the target correspondent, a notification of delivery is returned from the TELEX or FAX carrier that contains the actual cost of the transmission.

When FI correspondents do not have a test key relationship with other banks or institutions, they may send a message to requesting FI to validate a test result on their behalf. A third-party test of this sort results in a fee being charged back to the requesting correspondent. A daily report should provide a summary of all third-party test activity for the day, to facilitate the charging process.

2.1.11.3 Retention

FED and CHIPS messages are normally retained in the front-end systems for the current day only and queries regarding previous days' payments can be done through back-end applications or archival systems.

For SWIFT/TELEX/FAX, however, messages could be retained for a minimum of three business days, so that if an investigation shows that a message was misrouted or incompletely processed, an operator can recall the message to requeue or resend it.

2.1.12 Summary of the Requirements

The functional components are the fundamental building blocks of a financial messaging gateway. In an environment such as The FI, however, there is a higher level of functionality that must be addressed in order to ensure that the enormous volume of messages can be handled accurately and reliably every day. That higher level addresses the issues of scale, where performance, availability, configurability, flexibility, maintainability, visibility, and support are taken into consideration.

2.1.12.1 Performance and Availability

The following features are needed across all messaging components to assure that FI will be able to handle ever-increasing message volumes in a timely way, with minimal disruption:

- High throughput
- Infinite scalability
- Reliability and recoverability
- Continuous availability

2.1.12.2 Configurability, Flexibility, and Maintainability

Wherever possible, messaging components should provide a broad range of configuration options, and metadata-driven logic, so as to minimize the amount of code needed to customize behavior to suit FI requirements. Specific areas where metadata should be employed are:

- Transformation
- Validation
- Routing
- Workflow

In addition, FI requires that there be a rich and flexible user interface to manipulate messages as necessary in an emergency. This includes the ability to manually remove problem messages from a queue, and to manually place a message on a queue or at any transition point in a workflow.

2.1.12.3 Visibility

A mission-critical system such as the one that supports financial messaging must provide an operator with comprehensive, yet easy to read displays that show what's running normally and what is not. When problems do occur, an operator should be notified. The following features must be integrated across all components of the financial messaging infrastructure:

- Monitoring capabilities for devices, communications interfaces, processes, and workflow
- Integrated error reporting, logging, and audit trail
- Standards-based notification

2.1.12.4 Conclusion on the Requirements

There are a number of ways to achieve these "big picture" goals. One way is to adopt a set of standards that will be applied to all components in the

messaging infrastructure. The areas where these standards may apply are in problem notification, logging (to provide a centralized chronology across all components), message transport (i.e. WebSphere MQ), relational database, transaction managers, message headers and syntax, and possibly even programming language. Ideally, these standards are defined and maintained by third parties, and are not vendor-specific or proprietary. Some of these standards, such as programming language and relational database, are beneficial because of the efficiency that will be achieved in development, maintenance, and support. Others, such as notification and logging, will simplify and problem resolution.

In reviewing the functional features, it seems clear that no single product, or even a selected group of products, will give FI all the functionality that they currently have or are likely to want in the future. As a result, the "buy v. build" question may well be answered with "buy and build", that is, buy products that provide some core functionality, and tailor them to the specific needs of FI. The ease with which purchased products can be customized and enhanced, as well as the extent to which products are configurable or metadata-driven should be major factors in the product selection process.

2.1.13 Flowchart Examples

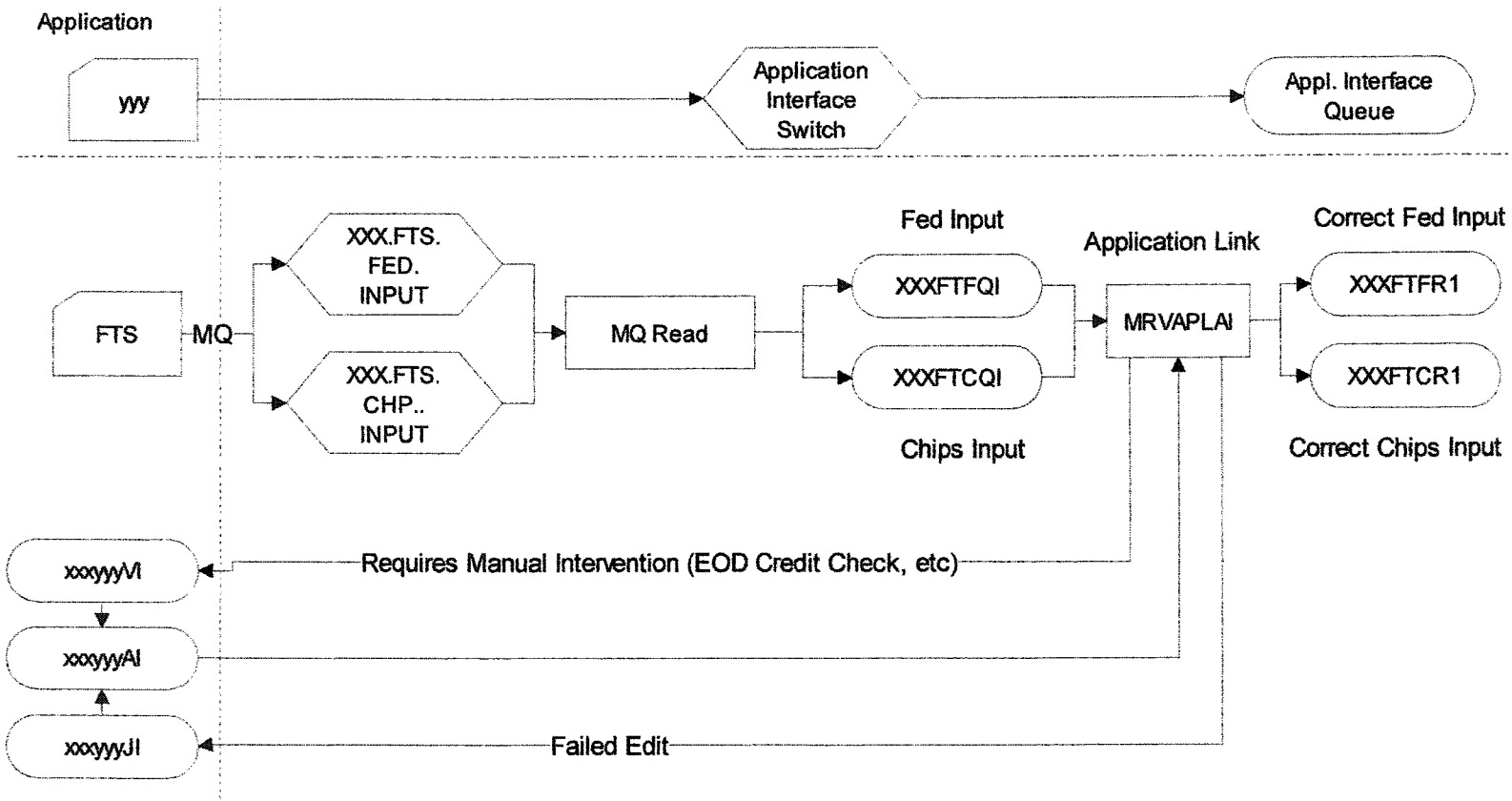


Figure 15 Input to FedWire/CHIPS (Fragment 1).

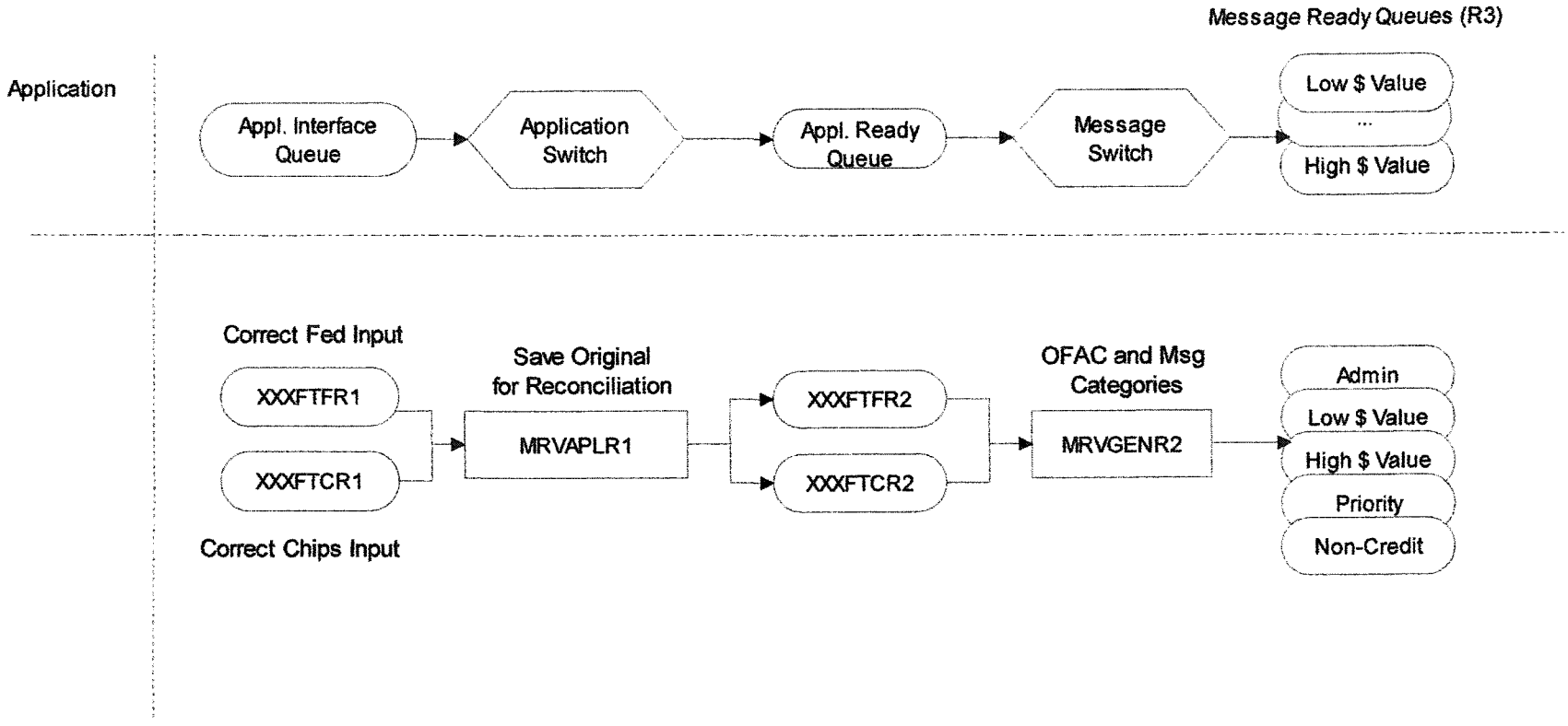


Figure 16 Input to FedWire/CHIPS (Fragment 2).

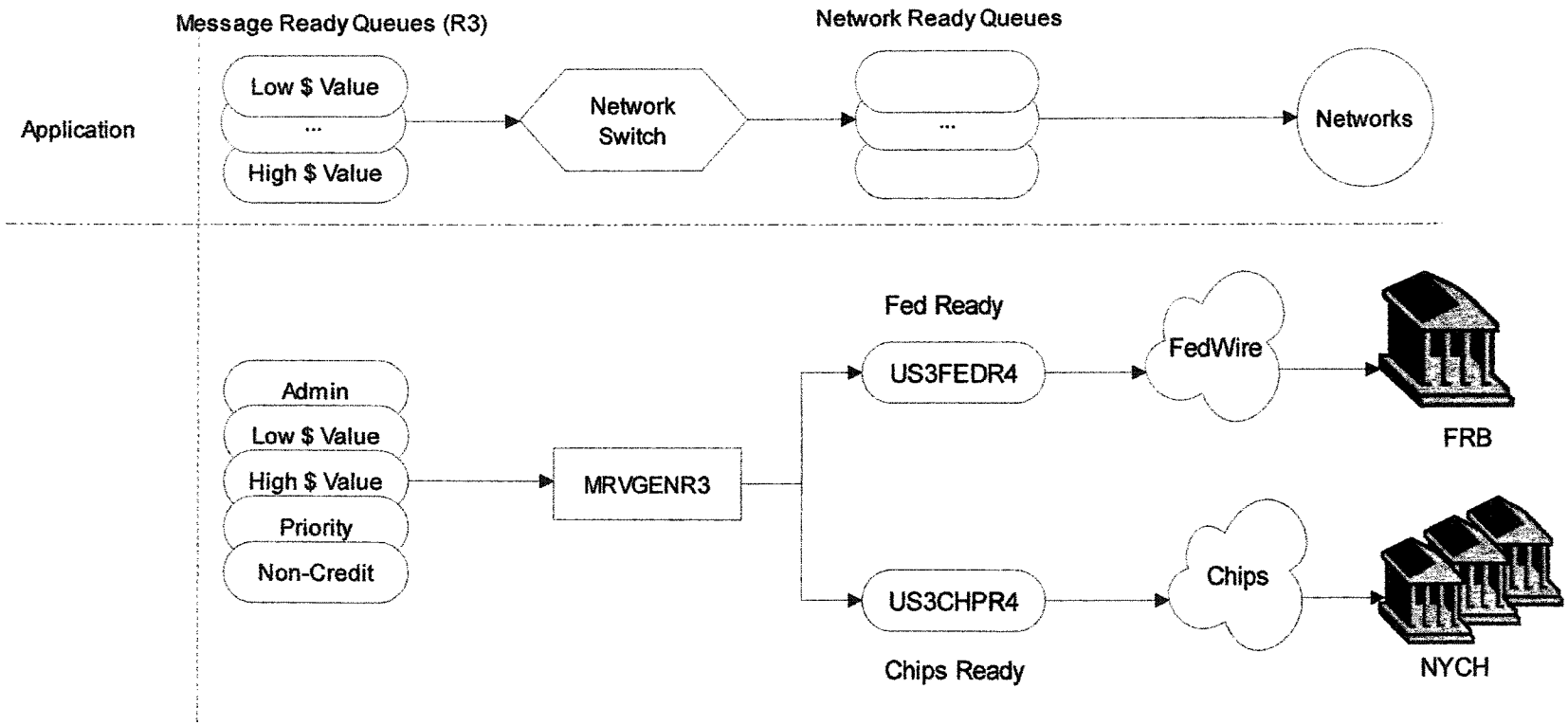


Figure 17 Input to FedWire/CHIPS (Fragment 3).

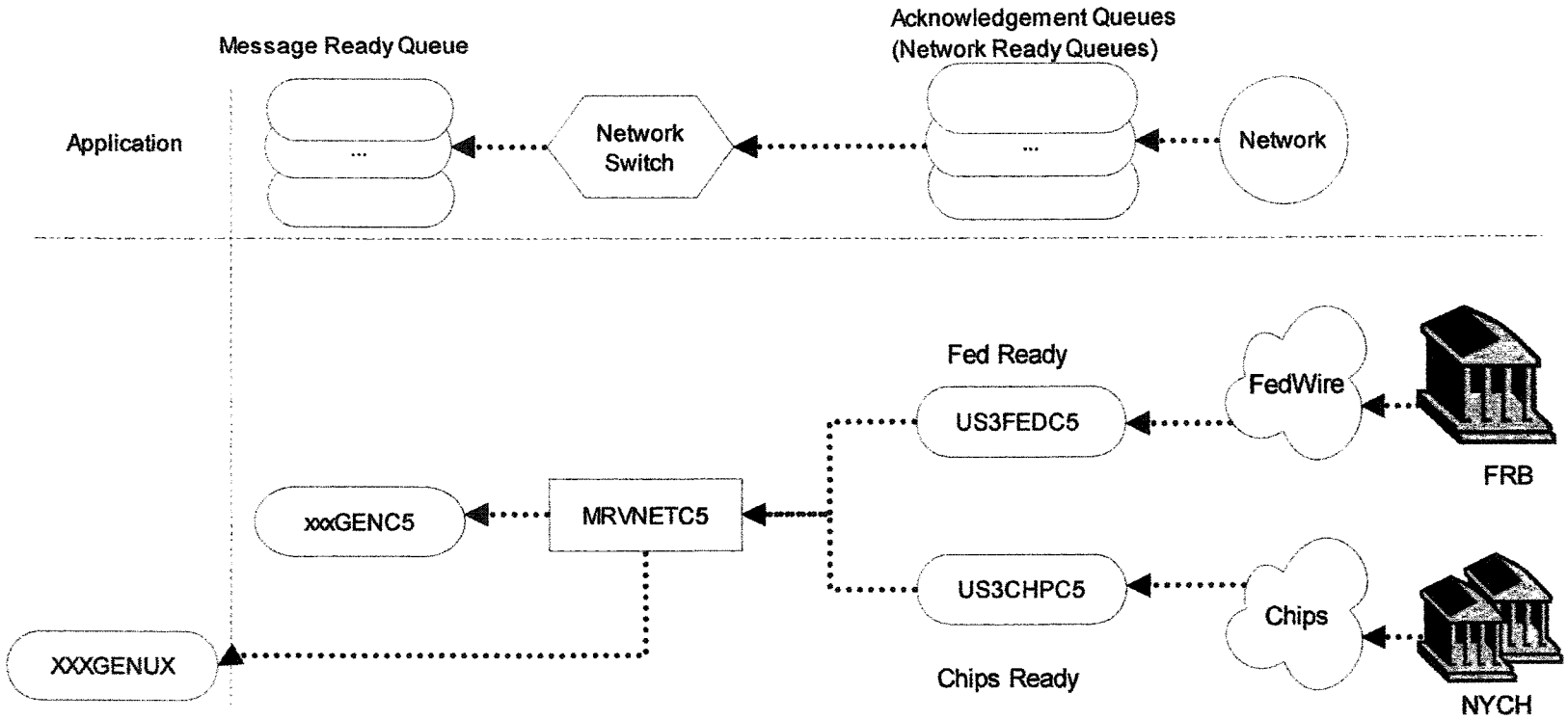


Figure 18 Acknowledgement on Input to FedWire/CHIPS (Fragment 1).

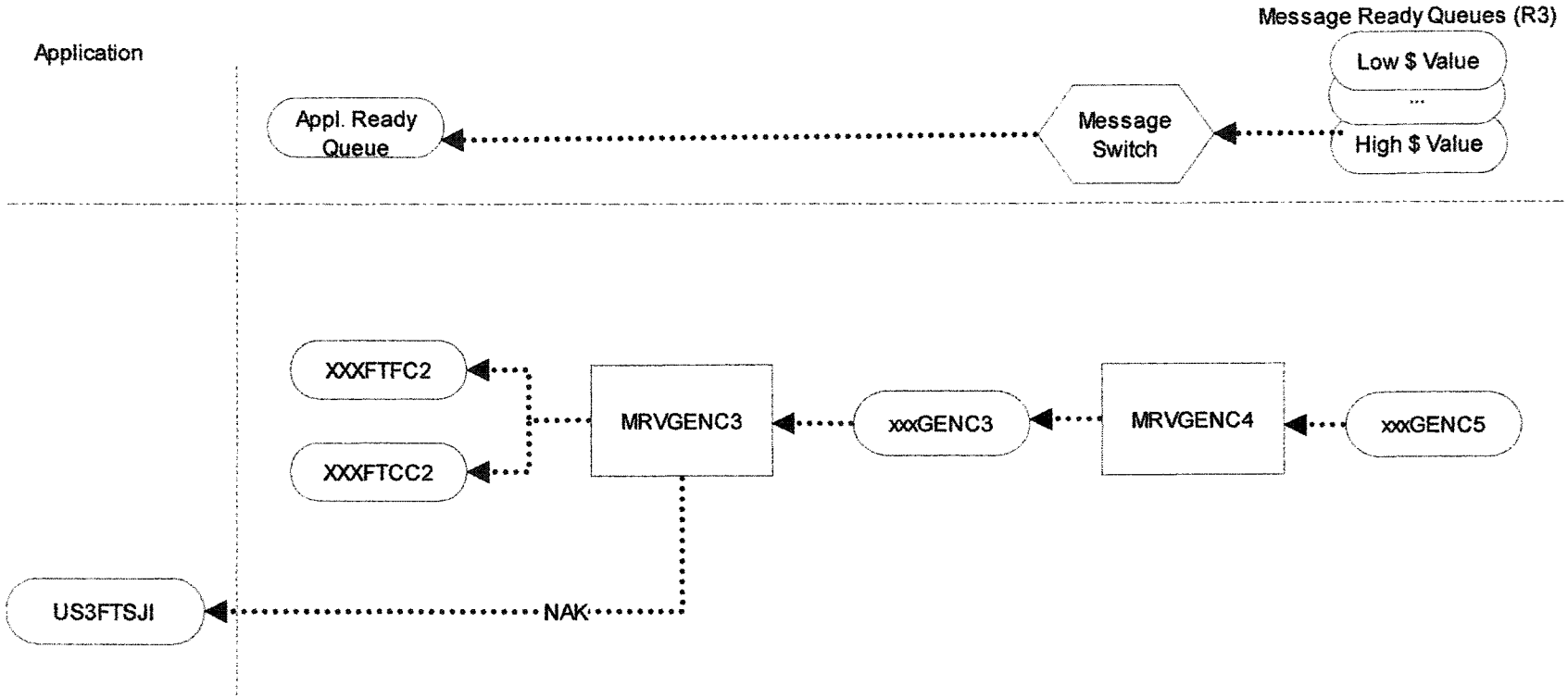


Figure 19 Acknowledgement on Input to FedWire/CHIPS (Fragment 2).

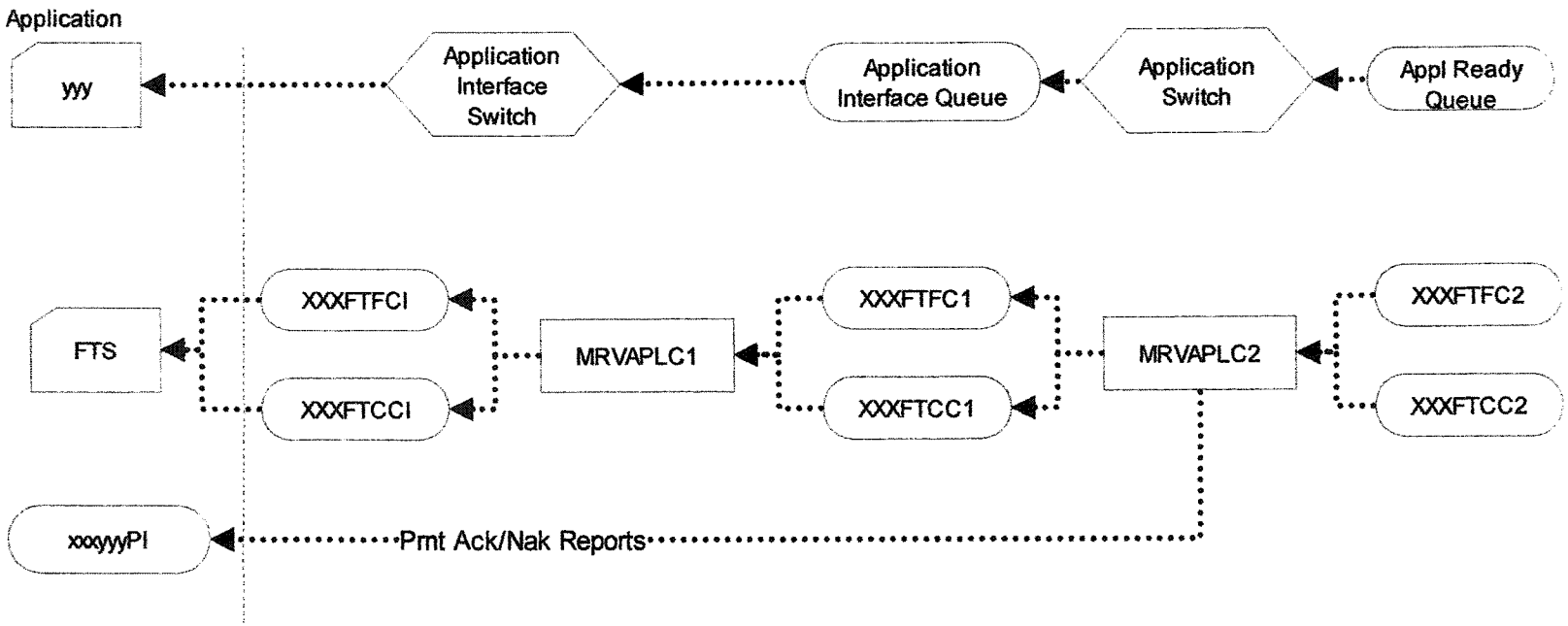


Figure 20 Acknowledgement on Input to FedWire/CHIPS (Fragment 3).

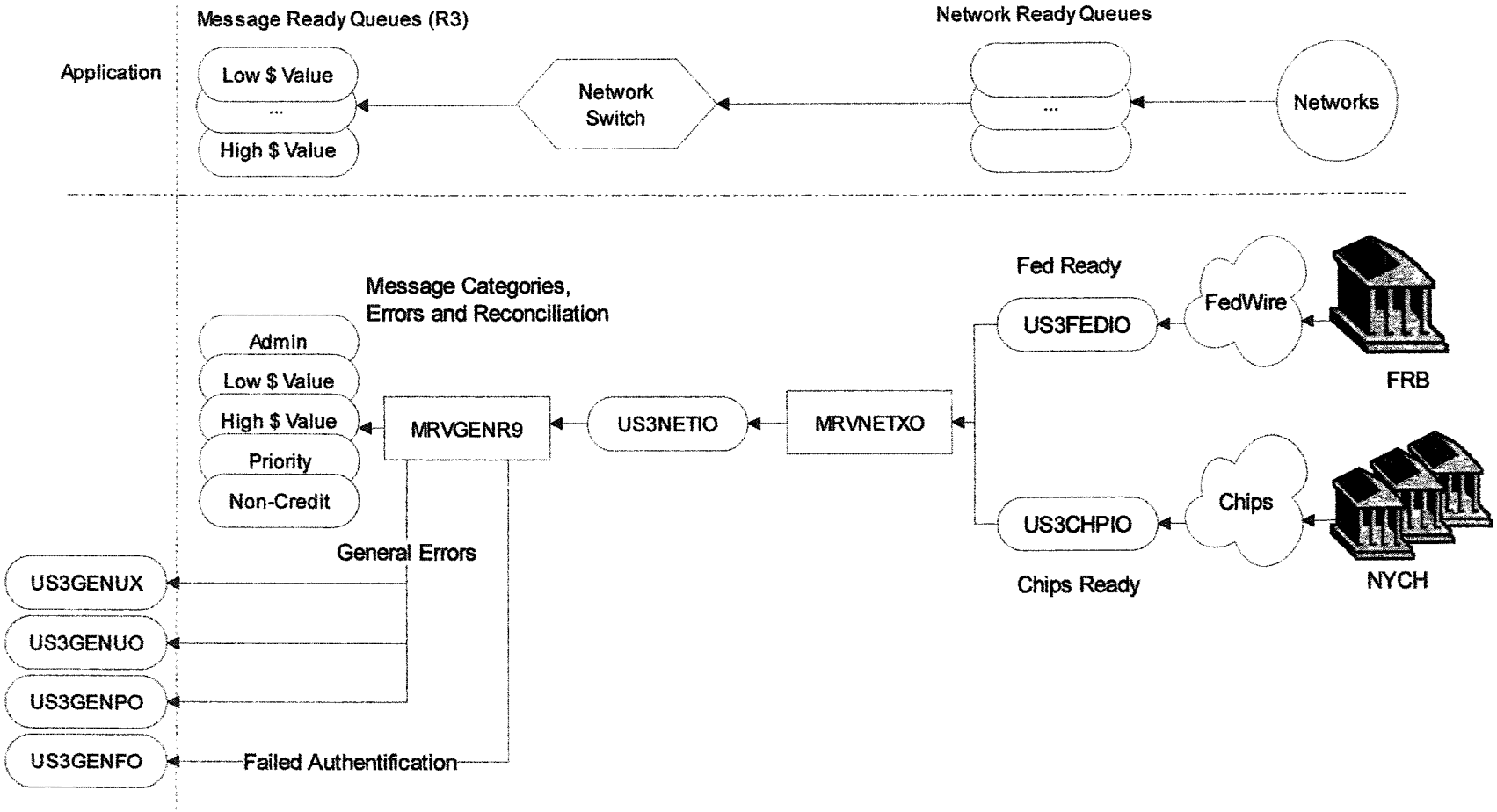


Figure 21 Output from FedWire/CHIPS (Fragment 1).

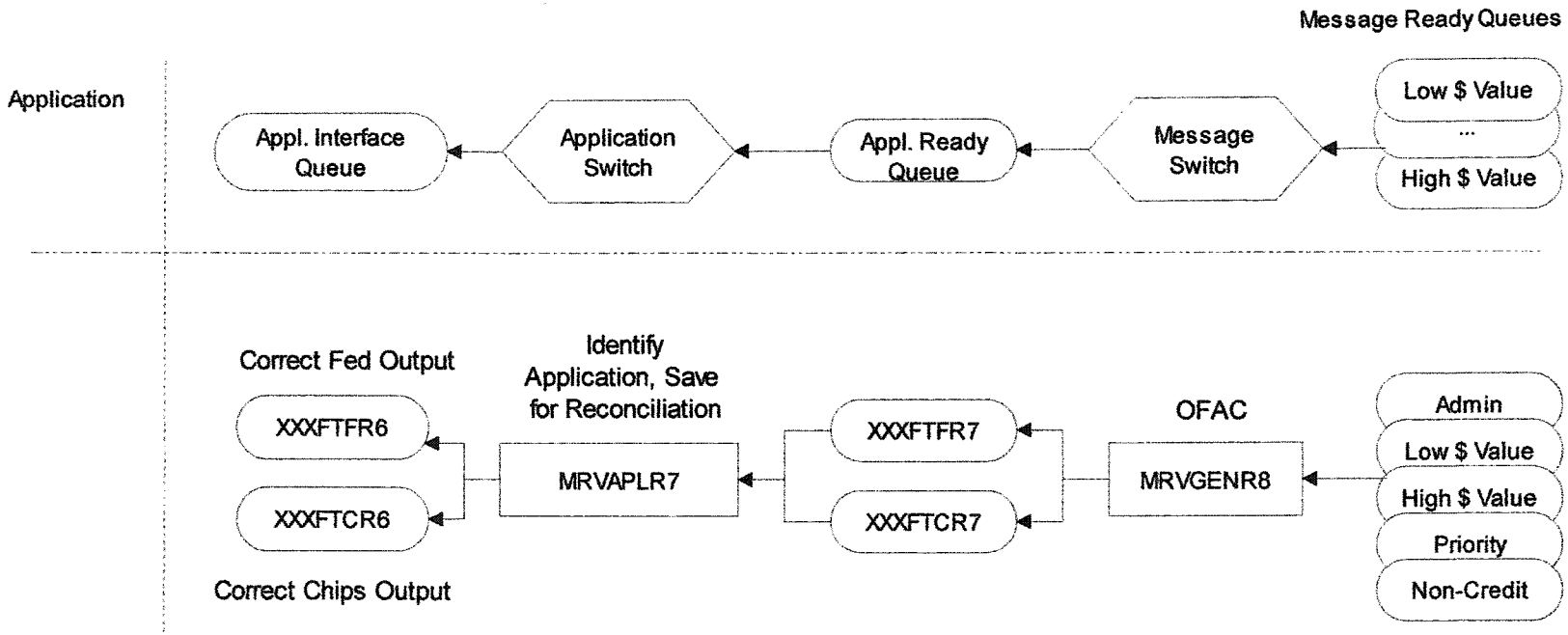


Figure 22 Output from FedWire/CHIPS (Fragment 2).

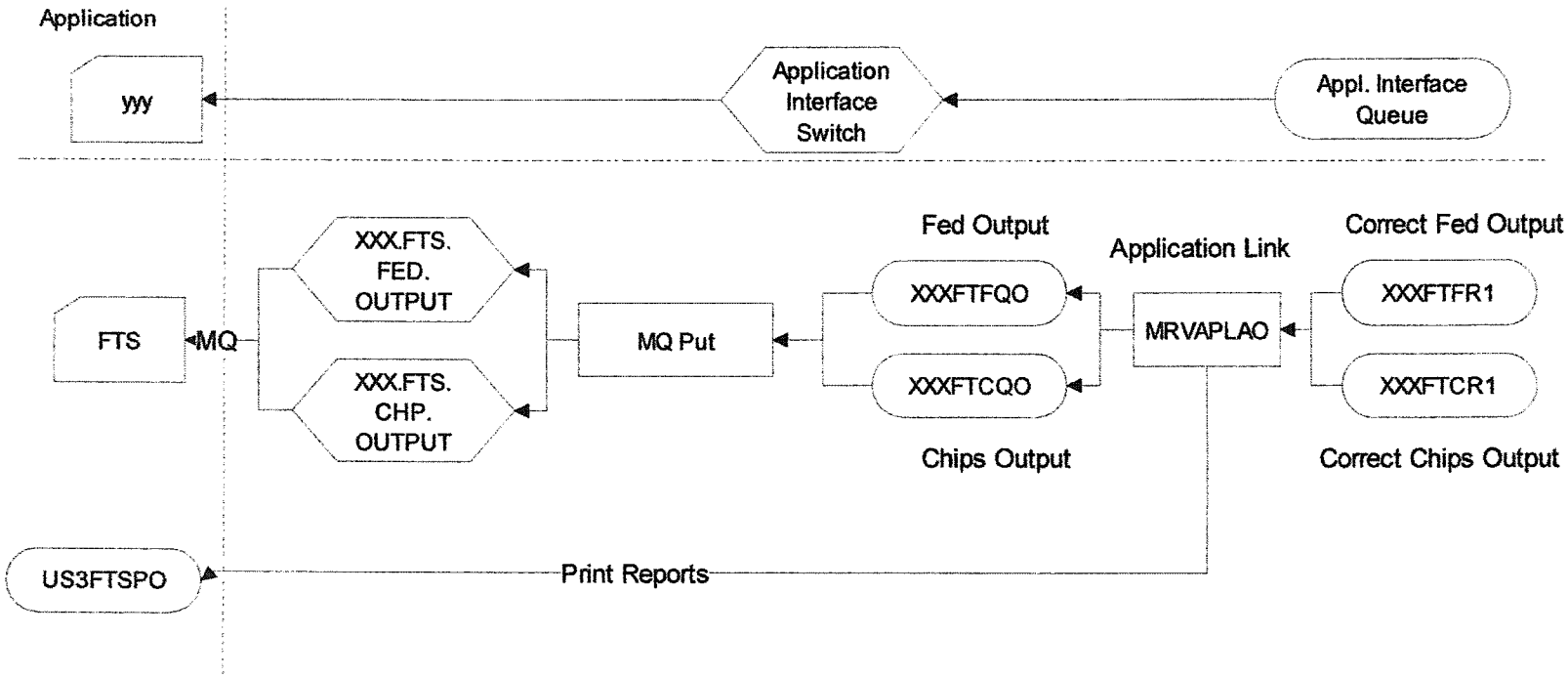
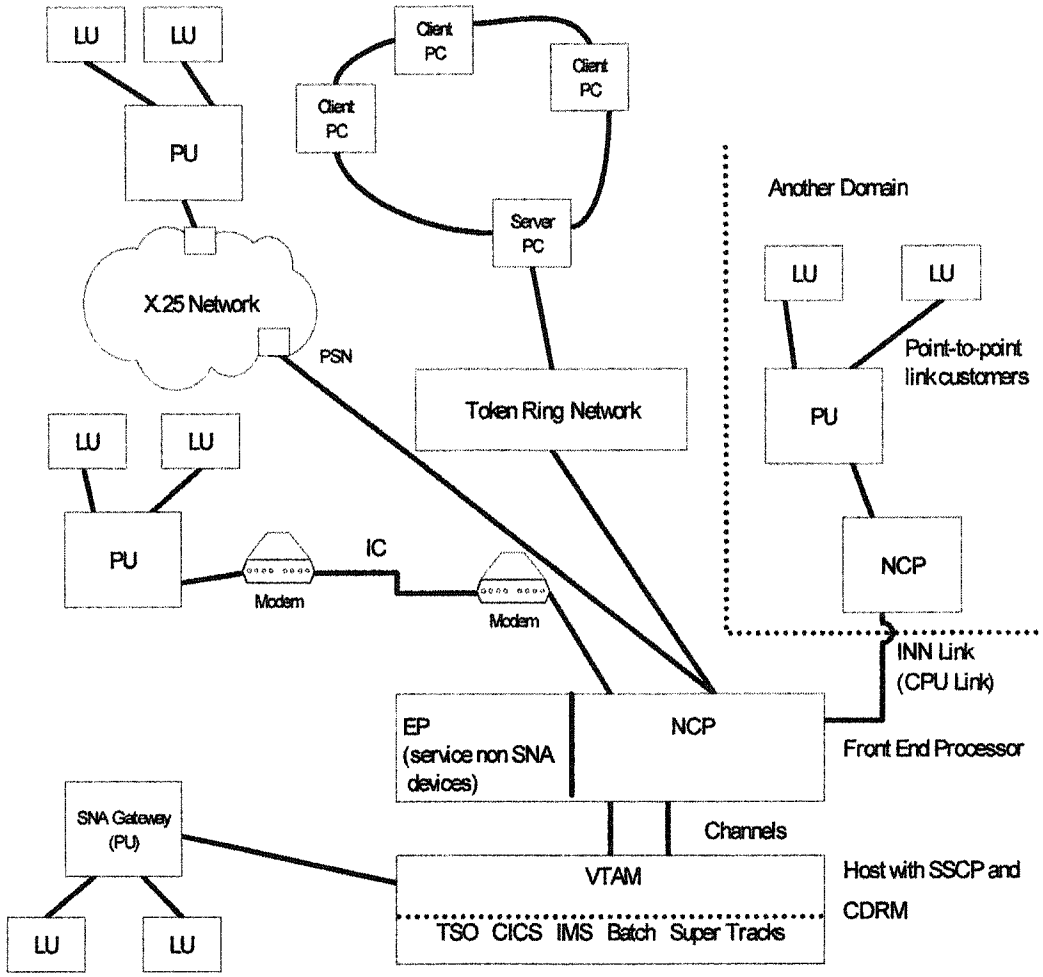


Figure 23 Output from FedWire/CHIPS (Fragment 3).

2.2 Building Blocks of Financial Institution Intelligent Network

2.2.1 Financial Institution Intelligent Network as Intelligent Peripheral

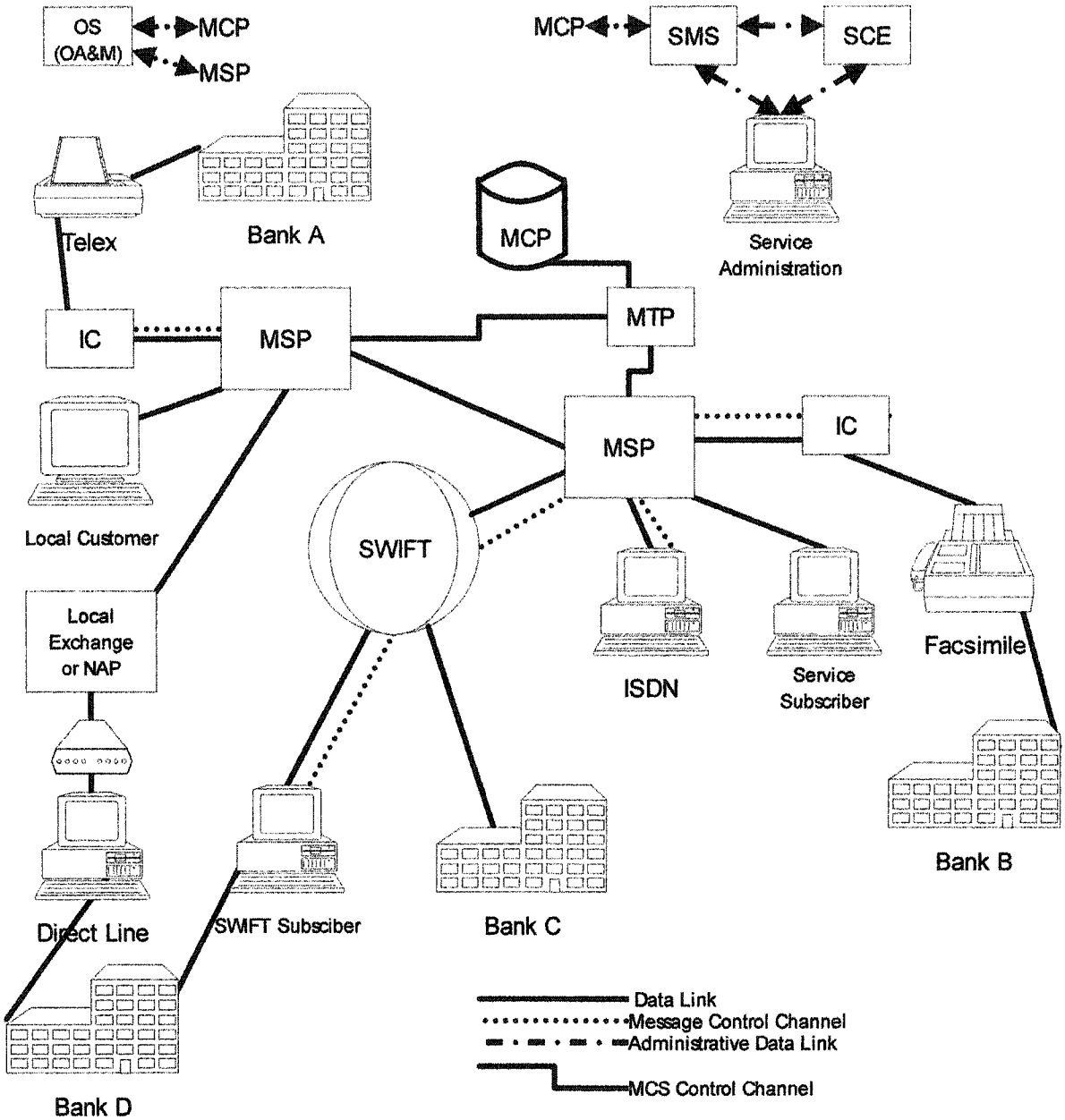
Most current Financial Institution Networks (FIN) are not truly intelligent though they have some routing intelligence embedded in SSP-like software. Example of worldwide FIN built as computer network depicted on Figure 24. Functional blocks of IN are not clearly presented. Control information is not separated from the data.



VTAM = Virtual Telecommunication Access Method; NCP = Network Control Program;
SSCP = System Service Control Point; CDRM = Cross Domain Resource Manager;
INN = Internal Network Node; EP = Emulation Program; IC = Interexchange Carrier;
PSN = Packet Switched Network; PU = Physical Unit; LU = Logical Unit;
CPU = Central Processing Unit; TSO = Time Sharing Option;
CICS = Customer Information Control System; IMS = Information Management System

Figure 24 Multiplatform Computer Network.

The new architecture of Financial Institution Intelligent Network (FIIN) is presented on Figure 25. FIIN can be considered as Intelligent Peripheral (IP) in Advanced Intelligent Network (AIN) architecture.



OS = Operating System; OA&M = Operations, Administration and Maintenance; MCP = Message Control Point; MTP = Message Transfer Point; MSP = Message Switching Point; MCS = Message Control System; SMS = Service Management System; SCE = Service Creation Environment; NAP = Network Access Point; ISDN = Integrated Services Digital Network; IC = Interexchange Carrier; SWIFT = Society for Worldwide Interbank Telecommunications.

Figure 25 Financial Institution Intelligent Network.

2.2.2 Bank Messaging System

BMS operates with messages as units of work or transmission objects. A message flow is a sequence of performed on a message by a series of message processing nodes. The actions can be defined in terms of the message format, its content, and the results of individual actions along the message flow.

BMS includes 4 switching layers within MSP where the message processing takes place. (IBM's WBI FN later defined message processing nodes, called primitives, that have some distant similarity to the switching layers).

BMS provides services for variable combinations of locations and applications. Bank Messaging System architecture is shown on Figure 26.

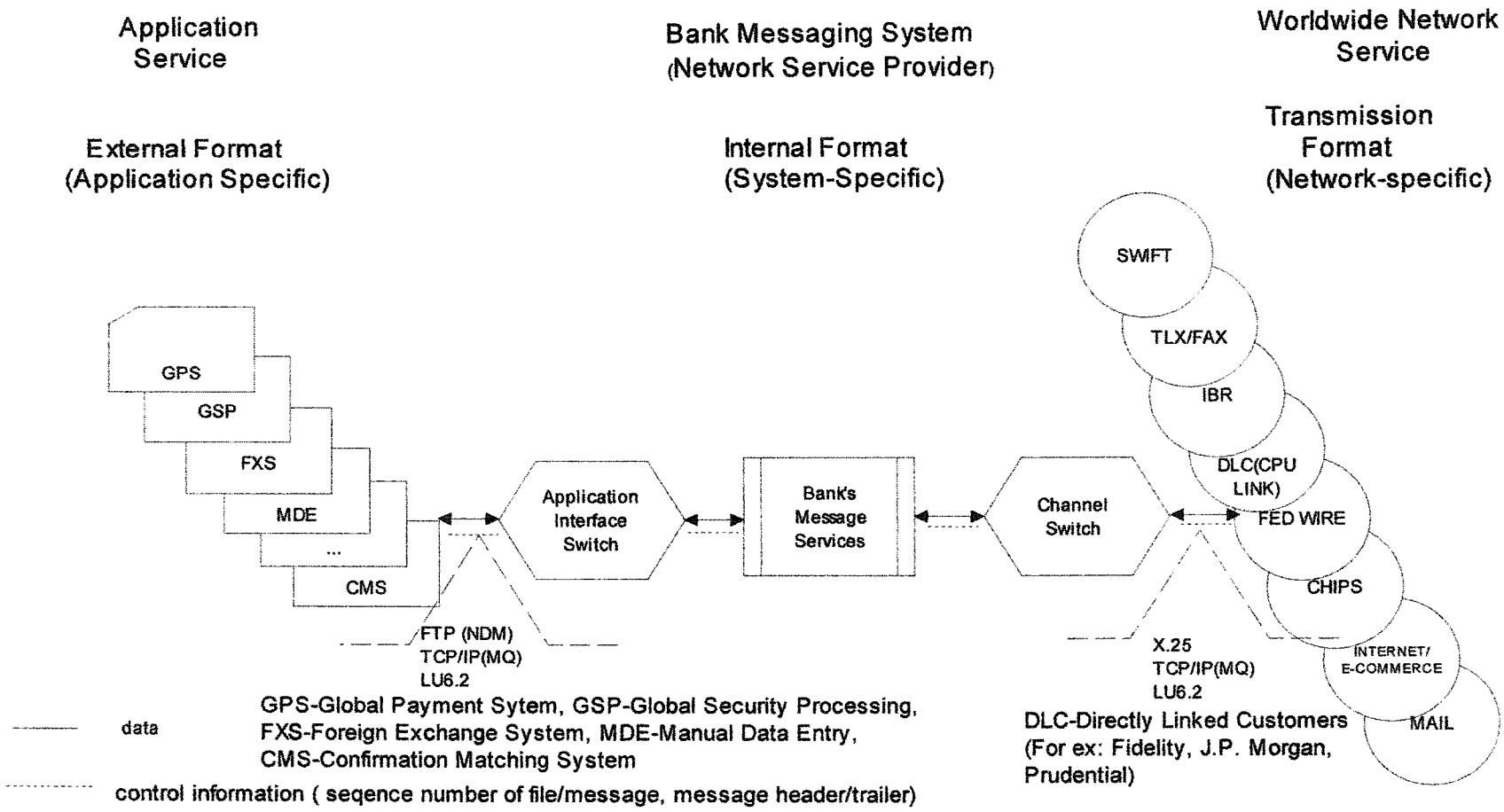


Figure 26 Bank Messaging System.

Typical applications interfacing with BMS are presented in Table 4. BMS applications for system operation and maintenance are listed in Table 5.

Abbreviation	Application Name
ADR	American Depository Receipts
ASR	Asset Reconciliation
BRK	Brokerage
CAN	Corporate Actions Notifications
CMS	Confirmation Matching System
DDA	Direct Deposit Account
GMC	Global Master Custody
GPS	Global Payment System
GSF	Global Short Term Investment Fund
GSP	Global Securities Processing
IFX	International Foreign Exchange for BEB
MDE	Message Data Entry
NOS	Nostro
REA	BMS Re-advising

Table 4 BMS Interfacing Applications.

Abbreviation	Application Name
0RE	BMS Reconciliation
APL	Application Routing
BKE	Bilateral Key Exchange
BRA	Branch and Application Specific
DLC	Direct Link Customers
ECO	E-Commerce Internet
FAX	Facsimile
GEN	Generic
GPA	General Purpose Application
IBR	Internal Bank Routing
MDE	Message Data Entry
MDL	Manual Delivery
MQI	Message Queue Interface
MSC	Miscellaneous
nSl	BMS System Module
OFA	OFAC for BMS
OFC	OFAC for BMS Users
SEQ	Sequencing
SLS	Secured Login Select
SWN	SWIFT Normal Priority
SWU	SWIFT Urgent
TLX	Telex
USE	User Security Enhancements

Table 5 BMS Applications.

2.2.3 Application Sub-layers

Application layer of OSI model is very specific to the type of processing being done. For message flow in Bank Messaging System application layer is divided into 4 sub-layers. As messages proceed through the BMS they are prepared to be ready for the next numbered step (see Figure 27) in 4 phases:

application interface,

application processing,

message processing,

network selection.

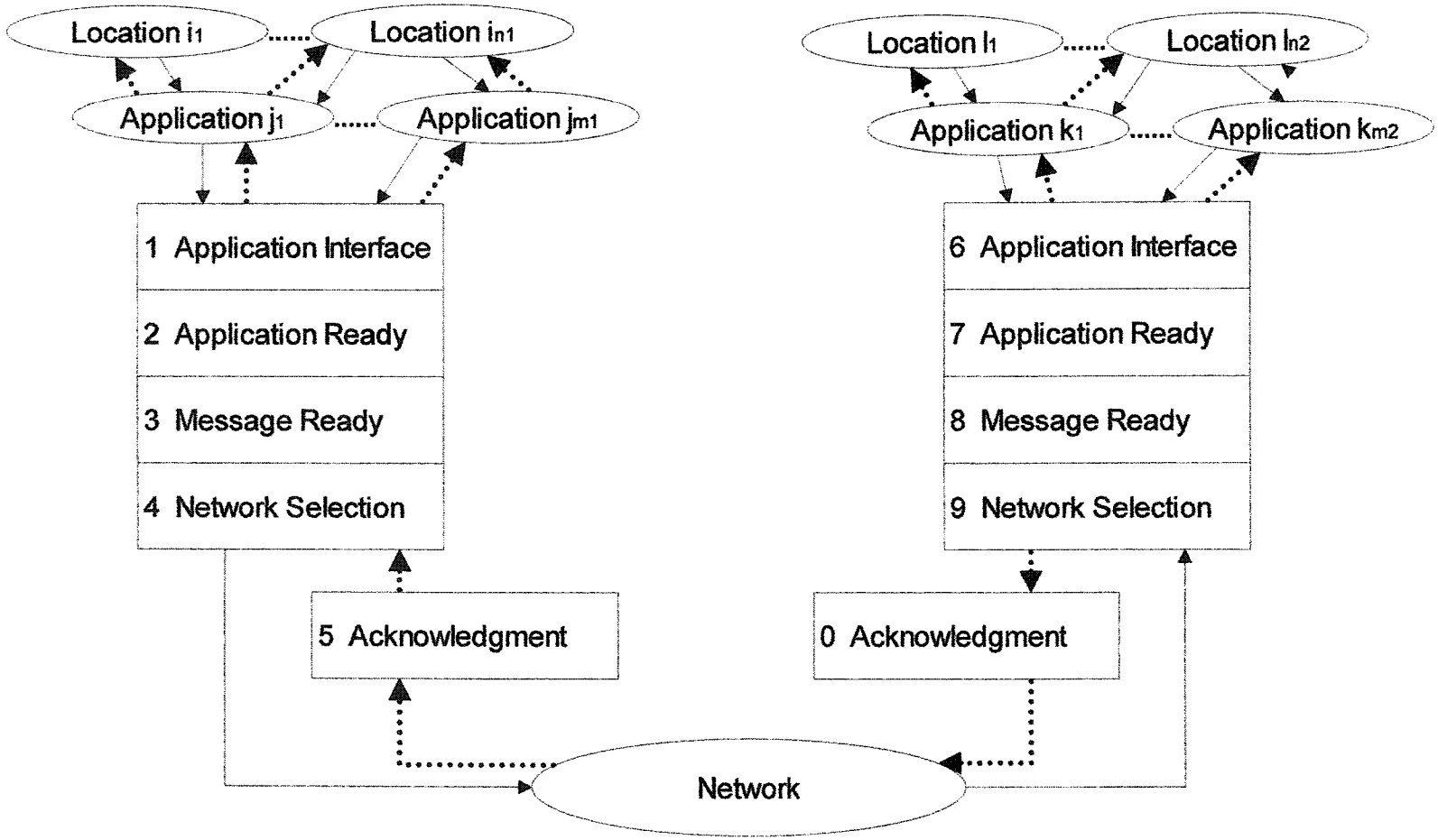


Figure 27 Application Sub-Layers.

2.2.4 Basic Building Blocks of Bank Messaging System

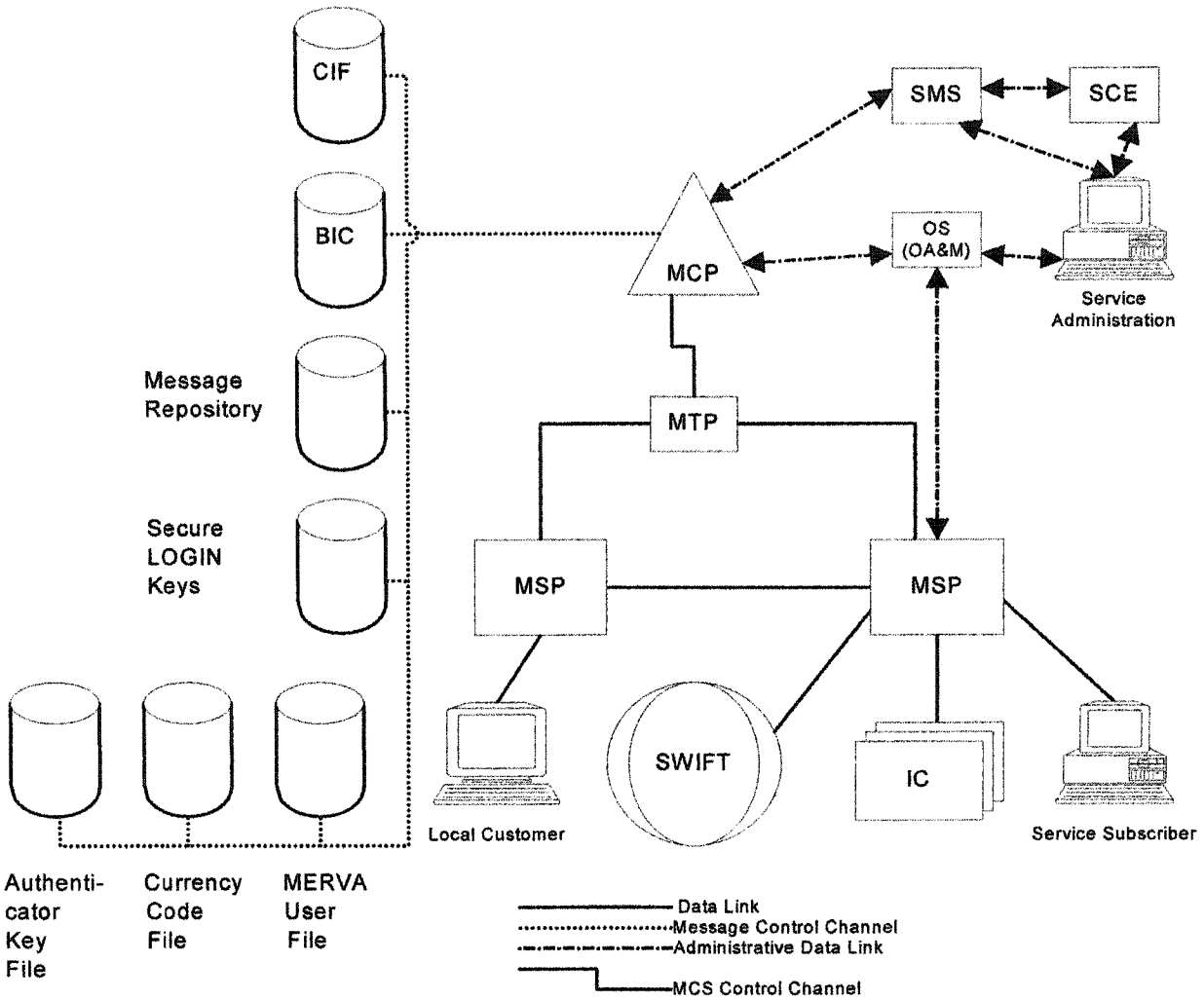
3 new functional components MSP, MTP, and MCP are introduced in architecture of FIN that mirror functions of similar components SSP, STP, and SCP of IN on message level. FIN connects MSP, MTP, and MCP by MCS links.

MSP provides access to FIN. All incoming and outgoing traffic goes through MSP. MSP links local subscribers, PVNs, ICs, other MSPs within FIN. MSP obtains service completion information from MCP and routes the messages correspondingly:

- local subscriber,
- private network connection for direct customers over dedicated lines (or CPU links),
- private carriers like SWIFT,
- IC like AT&T, MCI, WUI to send Telex,
- IC to send Facsimile,
- printer for mail delivery.

MTP provides communications between MSP and MCP via MCS links.

MCP is a virtual node within FIN containing systems and customer records databases presented on Figure 28.



OS = Operating System; OA&M = Operations, Administration and Maintenance; MCP = Message Control Point; MTP = Message Transfer Point; MSP = Message Switching Point; MCS = Message Control System; SMS = Service Management System; SCE = Service Creation Environment; IC = Interexchange Carrier; SWIFT = Society for Worldwide Interbank Telecommunications; CIF = Customer Information File; BIC = Bank Identification Code; MERVA = Message Entry and Routing for Various Applications.

Figure 28 Message Control Point.

MCP is a highly efficient parallel processor. Hardware components of MCP are similar to those of SCP:

- mass storage system with large capacity disk drives and disk controllers;
- parallel processors to communicate with I/O devices of the Data Bases;
- front-end processors to process queries received from MSP via MCS with predefined protocol (similar to SS7);
- front-end processors to communicate with SMS and OA&M.

Software components of MCP serve underlying hardware:

- data storage/retrieval system. Federal law requires to store financial messages (payments) for 7 years. For example, IBM product Report/Data Archive and Retrieval System (OnDemand) with Computer Output to Laser Disk (COLD);
- data base administration and maintenance including systems and customer records updates, backup and disaster/recovery functions;
- software interface with MSP via MTP;
- software interface with SMS and OA&M.

Some of the major functions of the MCP are the following:

- authentication of the sender;
- address translation of the receiver from customer identification (CID) or name key (look-up of CID or BIC);
- identification of a carrier authorized for message transfer.

SMS keeps systems and customer records on the data bases up to date. Systems records are FIN and message processing software for variety of applications.

2.3 Message Switching Point (MSP)

2.3.1 Four Layers of Switching

Fragment of Bank Messaging System (BMS) on Figure 29 displays four switching layers in message flow. Multiple BMS instances scenario is shown on Figure 30.

Message queuing is best suited for store-and-forward approach. Queue names can serve as parameters for underlying software and become crucial. Variant of queue name interpretation limited in length to 8 characters is presented in Table 6. Information describing the naming conventions on the charts is shown in Table 8 and Table 9.

BMS queue names are always 8 characters in length and interpreted as **brcaplfd** where:

br	bank's branch country code (see table 2)
c	city (see table 2)
apl	bank's applications mnemonic (see table 3) BMS applications mnemonic (see table 4)
f	function (see table 5)
d	direction (see table 6)

Table 6 Queue Name Interpretation.

Country Code	Country Name	City Code	City Name
Bank ABCD			
BE	Belgium	B	Brussels
DE	Germany	F	Frankfurt
GB	Great Britain (United Kingdom)	2	London
		A	London (Sub branch AMS)
HK	Hong Kong	H	Hong Kong
JP	Japan	J	Tokyo
		8	Osaka (Branch 848)
LU	Luxembourg	L	Luxembourg
SG	Singapore	S	Singapore
US	United States	3	New York
XX	Shared by more than one branch	X	Shared by more than one branch
Bank EFGH (Alternative NSP for Bank ABCD)			
GB	Great Britain	E	London
LU	Luxembourg	E	Luxembourg
XX	Shared by more than one branch	X	Shared by more than one branch

Table 7 Bank's Branch Country and City Codes.

Function	Description
A	Authorization
B	Backup
C	Acknowledgement
D	Batch Data (ex NDM)
E	Data Entry
F	Failed (authentication), Function (table)
G	Control, GPA
H	Hold
I	Intermediate
J	Rejection
K	Backup (second)
L	Log
M	Message type
N	Normal Priority
O	Confirmation on Delivery
P	Print, Process
Q	Online Data (ex. MQ)
R	Ready
S	Send
T	Template (form), Telex (message category)
U	Urgent Priority
V	Verification
W	Confirmation on Arrival
X	Shared
Y	Received on MQ
Z	Reply on MQ
0	Message Category 0, System
1	Message Category 1, Print
2	Message Category 2, Print
3	Message Category 3, Print
4	Message Category 4, Print
5	Message Category 5, System
6	Message Category 6, Print
7	Message Category 7, Print
8	Message Category 8, Print
9	Message Category 9, Print
$	System

Table 8 Queue Functions.

Direction	Description
1,6	Queue level 1
2,7	Queue level 2
3,8	Queue level 3
4,9	Queue level 4
I,1,2,3,4	I type, Input to Network Queue Level
O,6,7,8,9	O type, Output from Network Queue Level

Table 9 Queue Direction.

Module Names are BMSxbbfl where:

xbb	Application mnemonic (see table 3 and table 4). If x is a letter, xbb is the 3 character application mnemonic. If x is a digit, bb is the first and second characters of application mnemonic.
f	Function
l	Processing level (0 to 9); I,O; T-table

Table 10 BMS Module Names.

First Character	Module Type
0	Cobol
1	Assembler,Definition table
2	Routing table
3	Linkcard
4	Macro
5	Copy
a	First character in application mnemonic

Table 11 BMS Module Types.

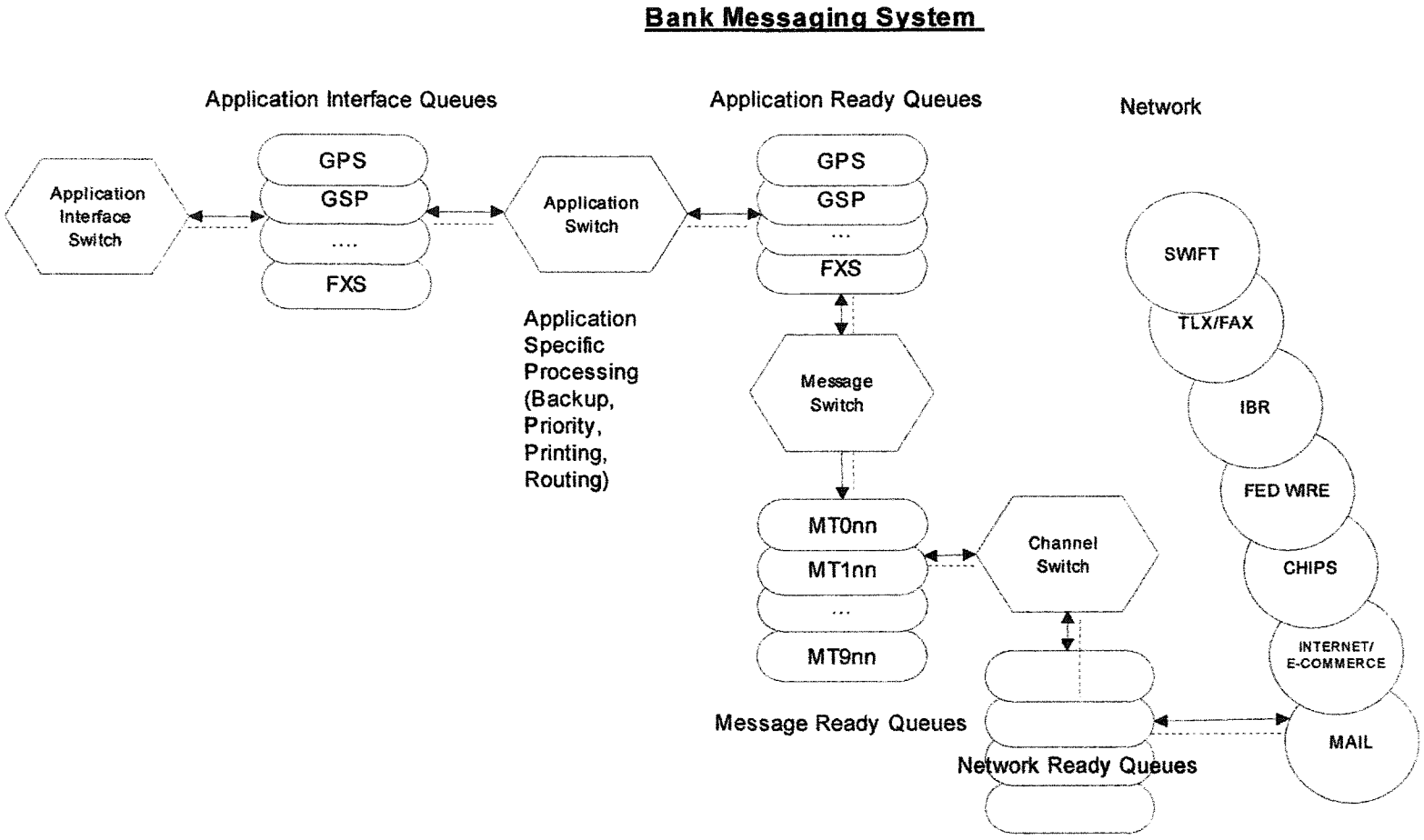


Figure 29 Four Layers of Switching.

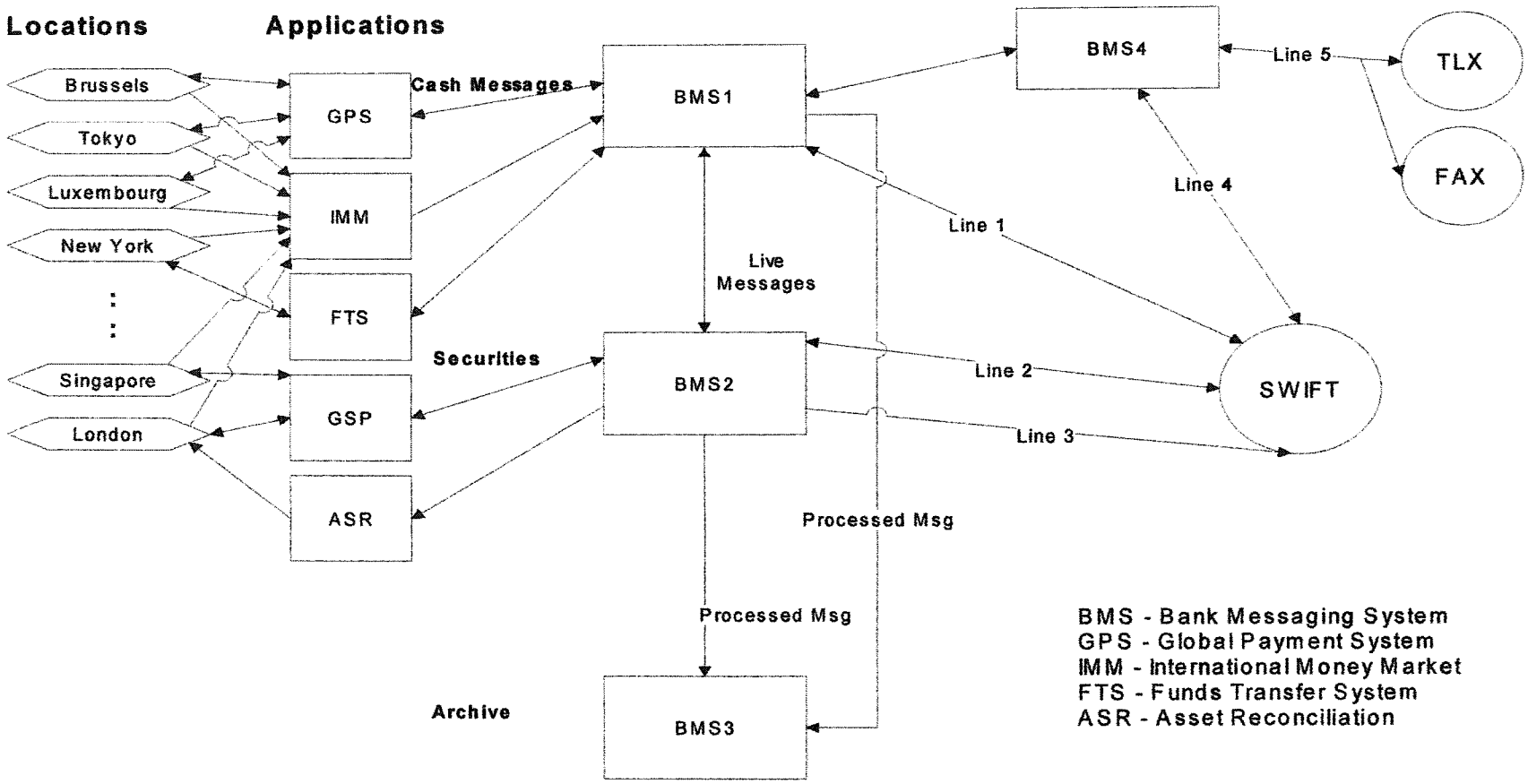


Figure 30 BMS Multiple Instances Scenario.

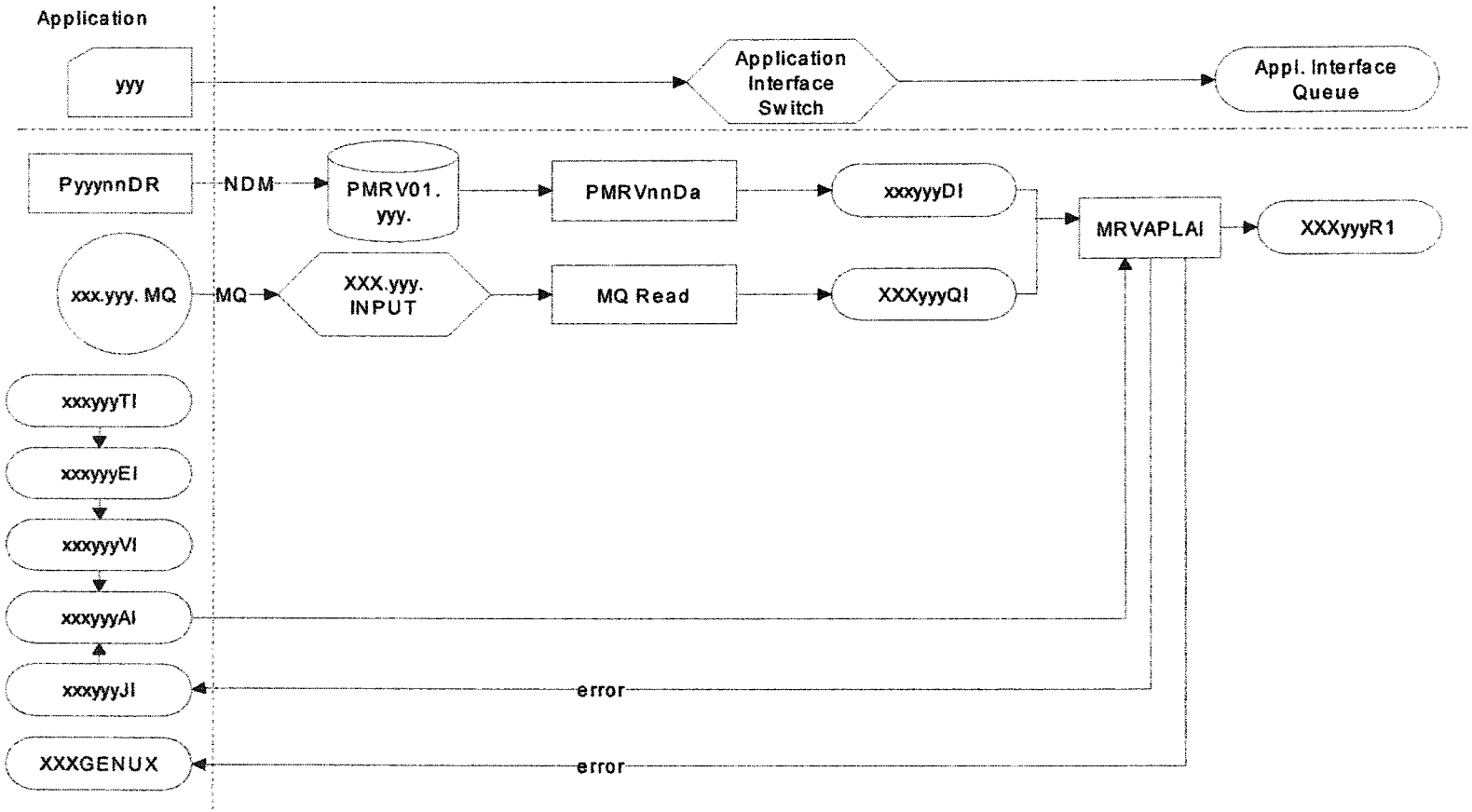


Figure 31 Input to Network (Fragment 1).

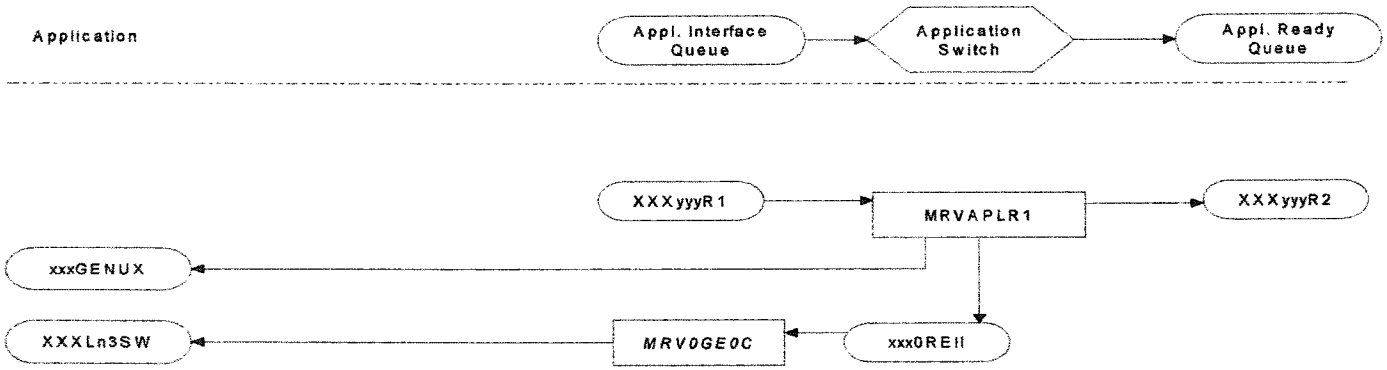


Figure 32 Input to Network (Fragment 2).

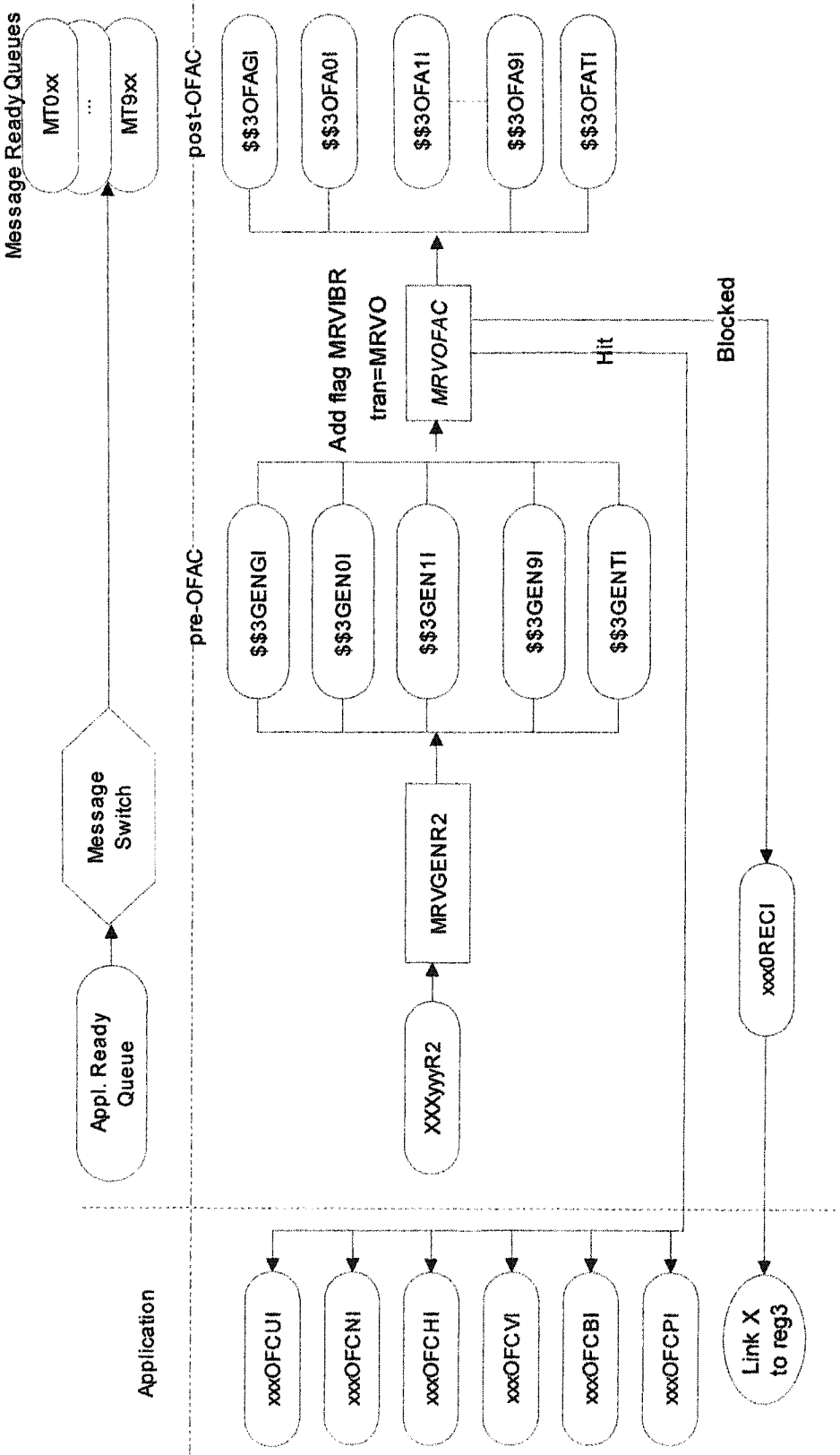


Figure 33 Input to Network (Fragment 3).

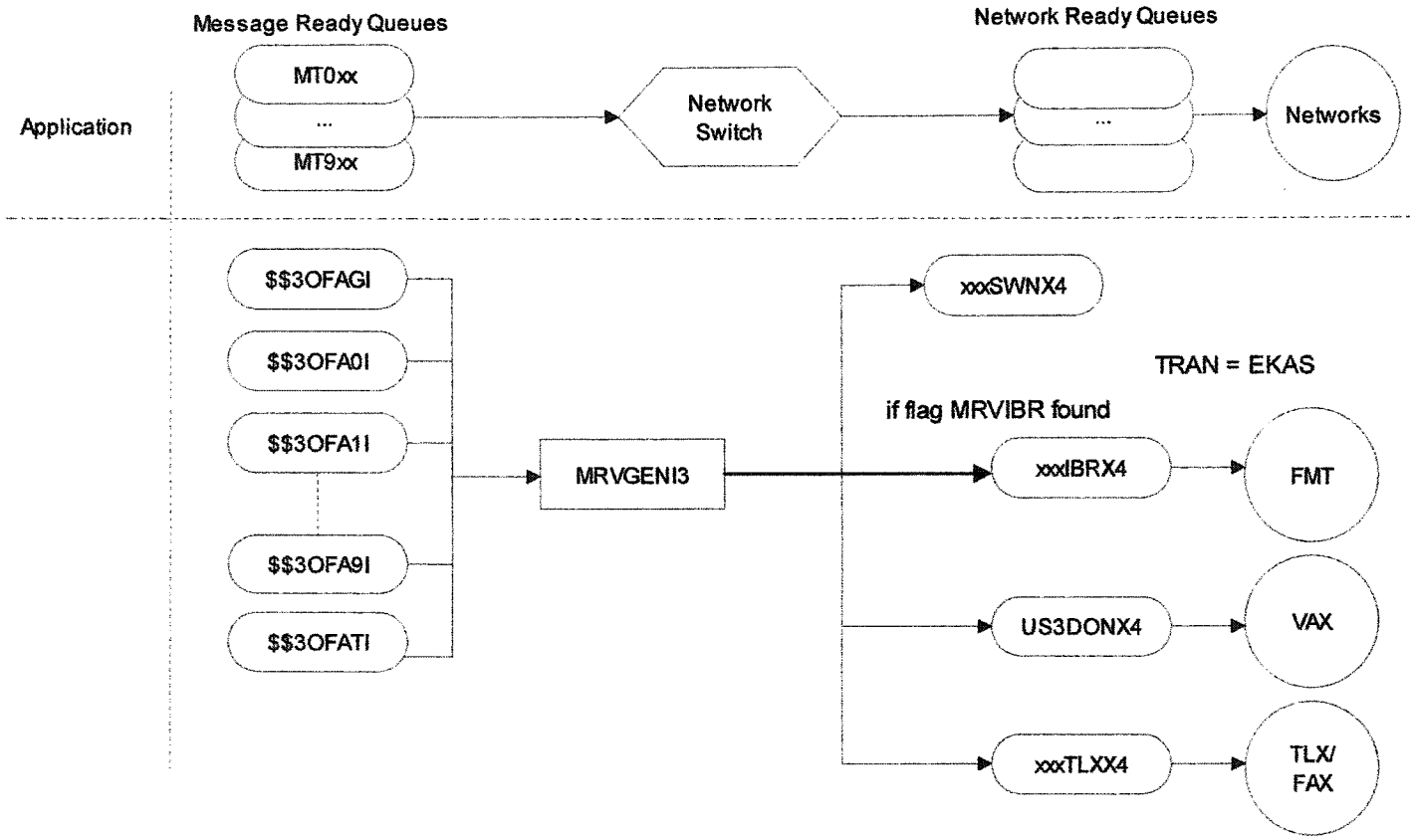


Figure 34 Input to Network (Fragment 4).

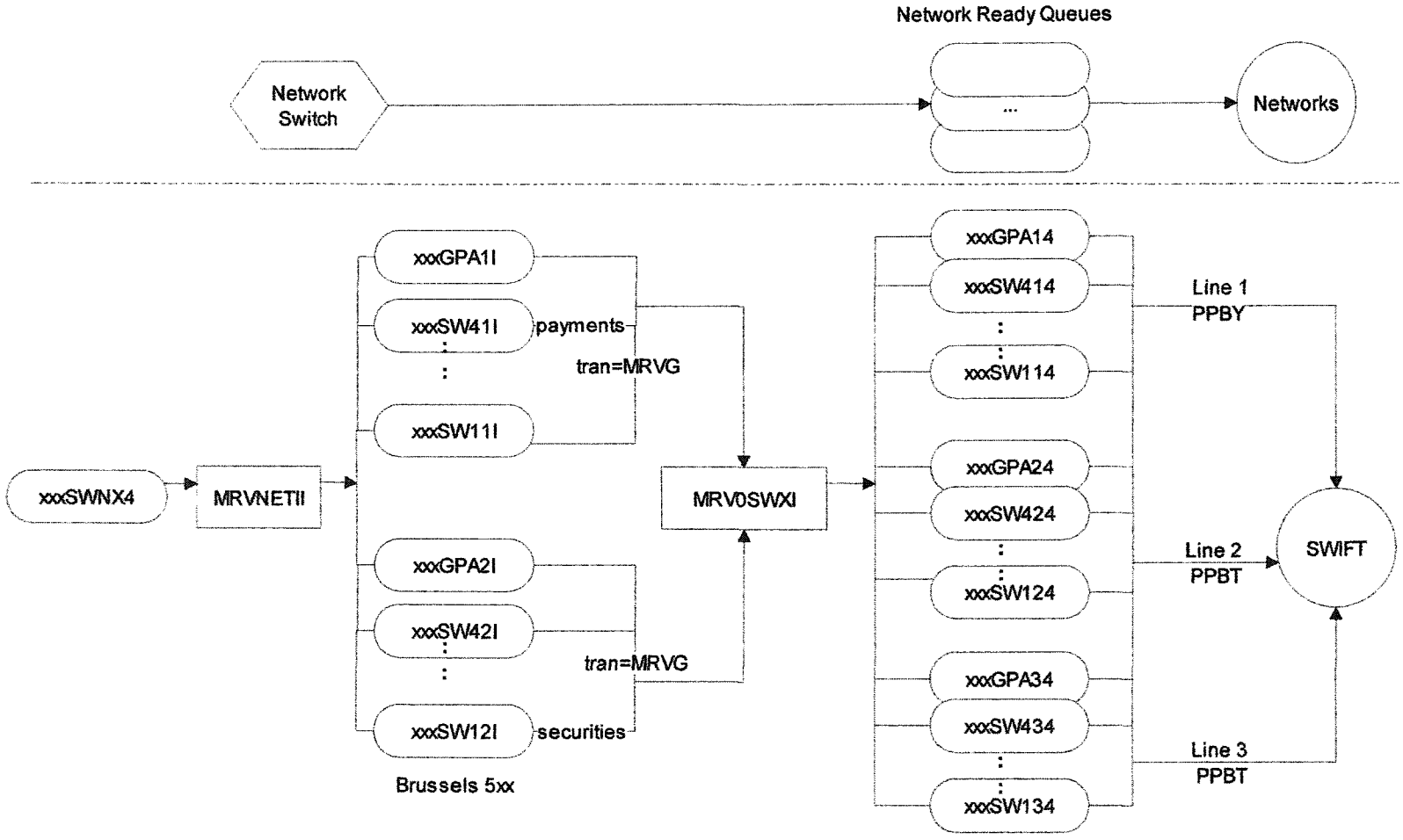


Figure 35 Input to Network (Fragment 5).

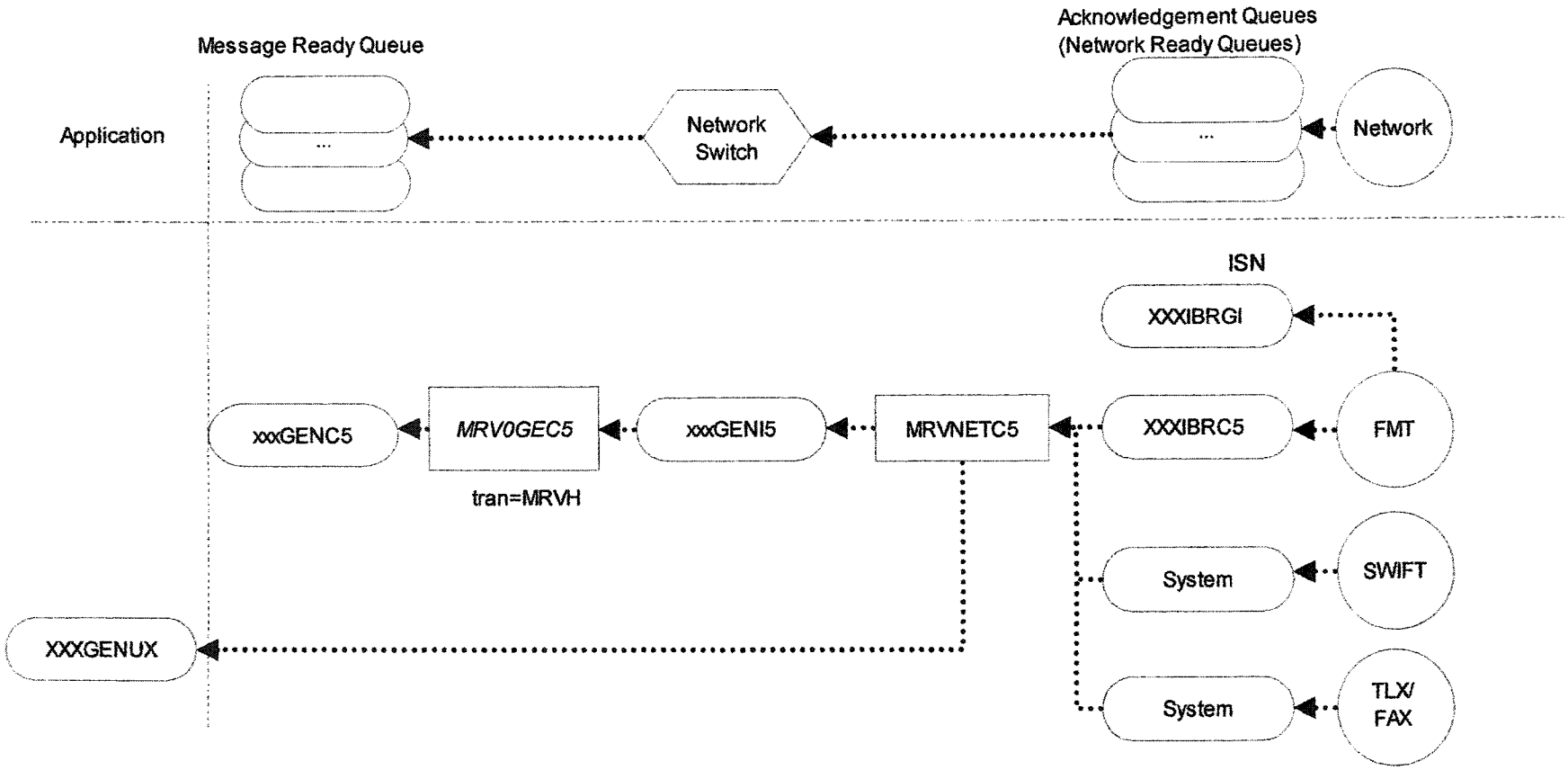


Figure 36 Acknowledgments on Input to Network (Fragment 1).

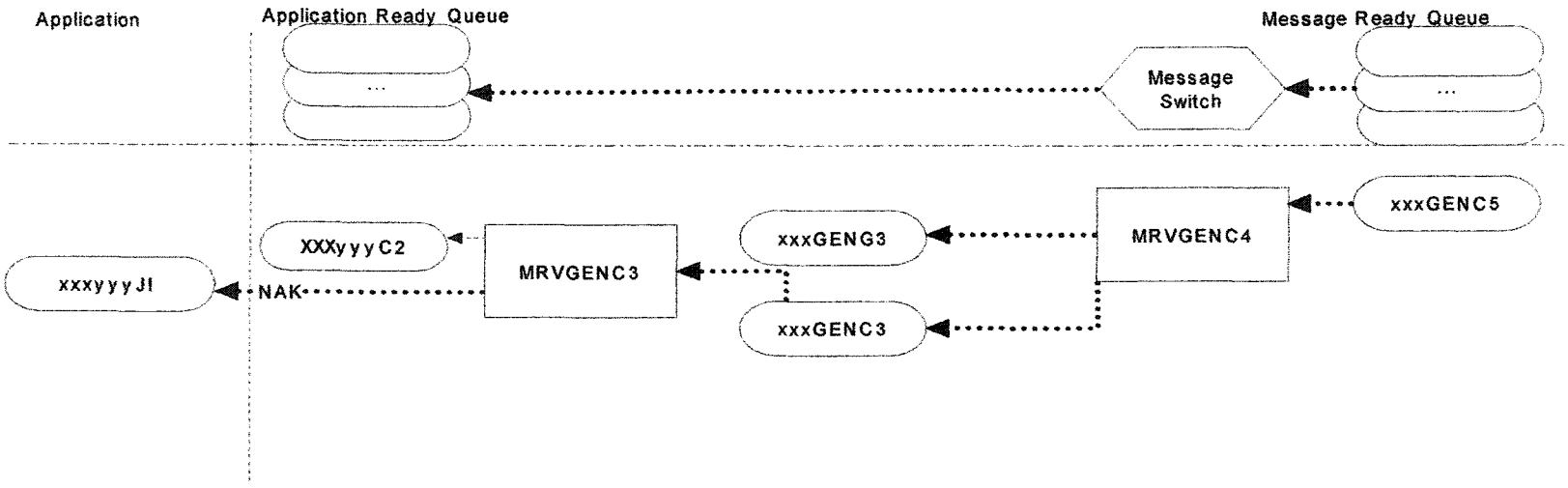


Figure 37 Acknowledgments on Input to Network (Fragment 2).

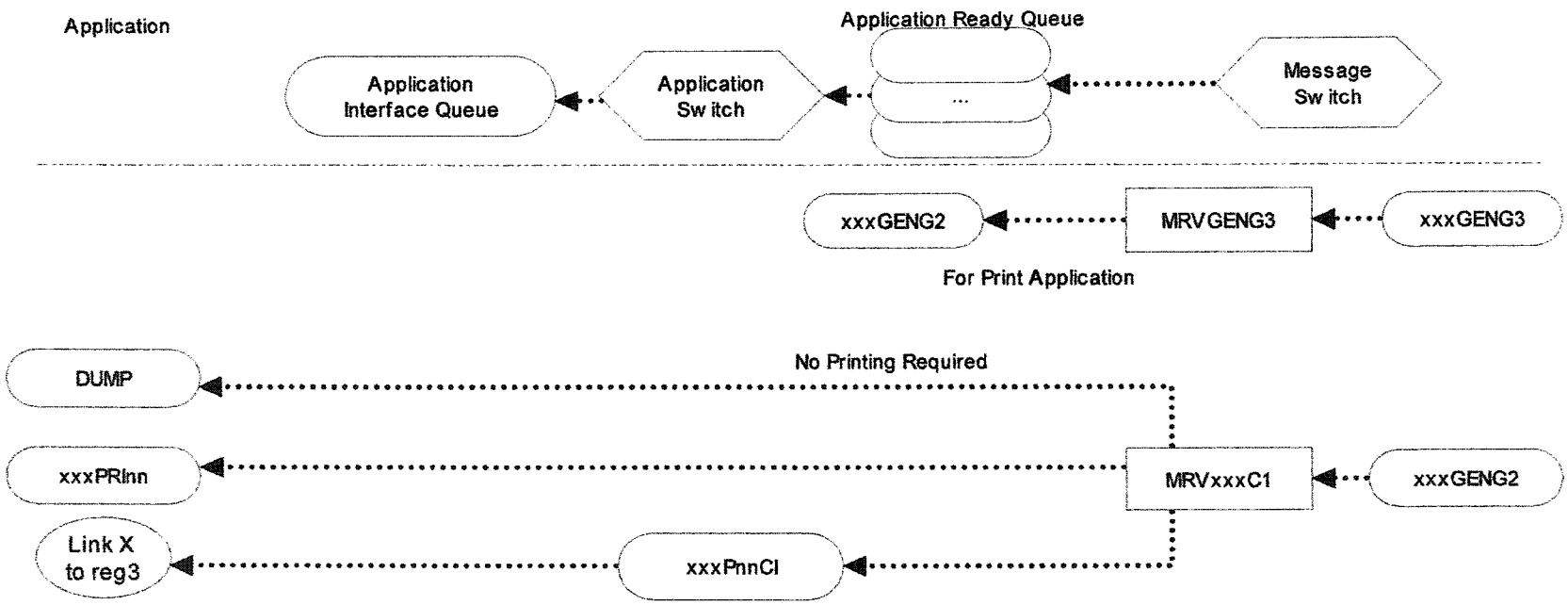


Figure 38 Acknowledgments on Input to Network (Fragment 3).

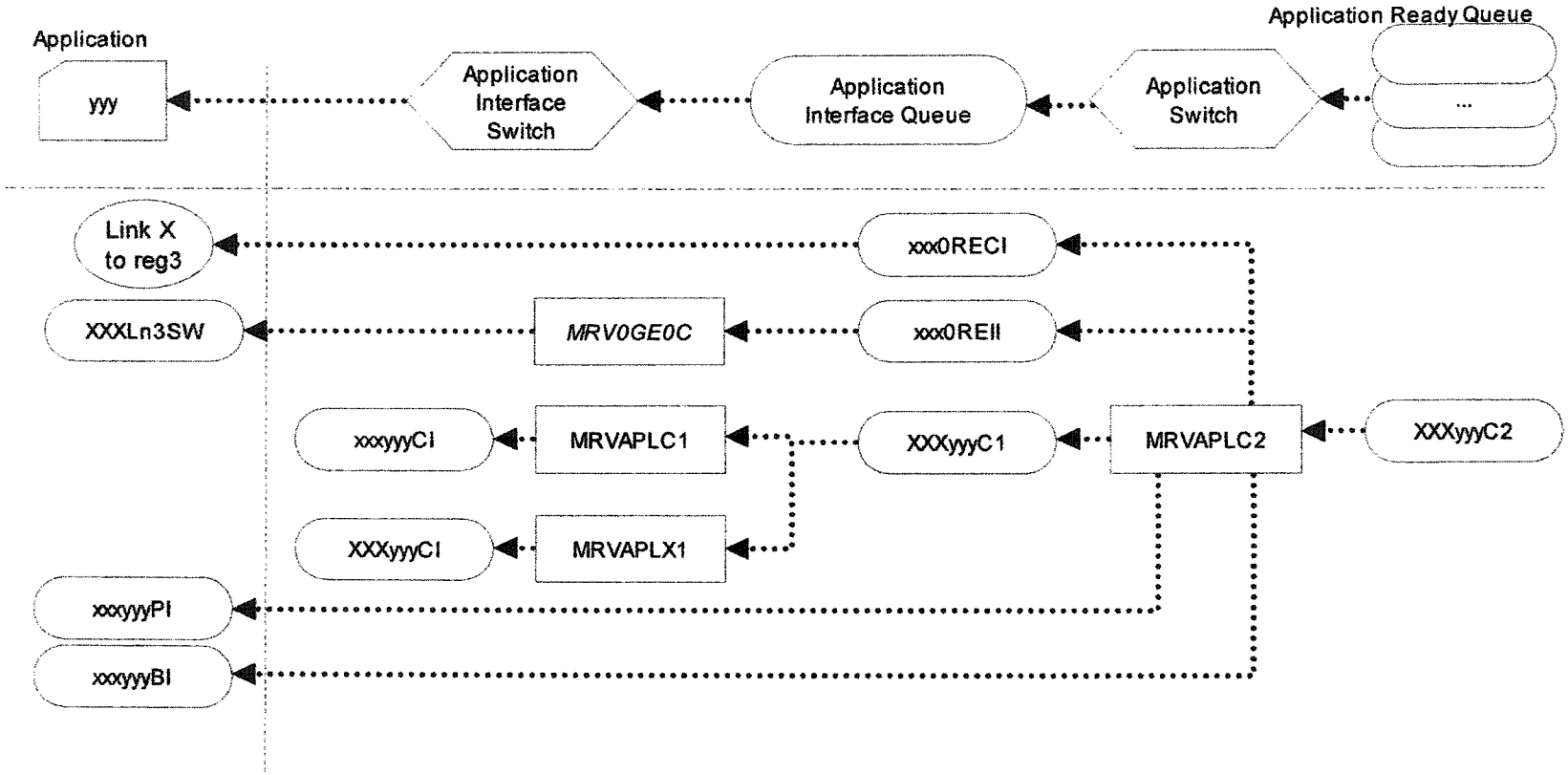


Figure 39 Acknowledgments on Input to Network (Fragment 4).

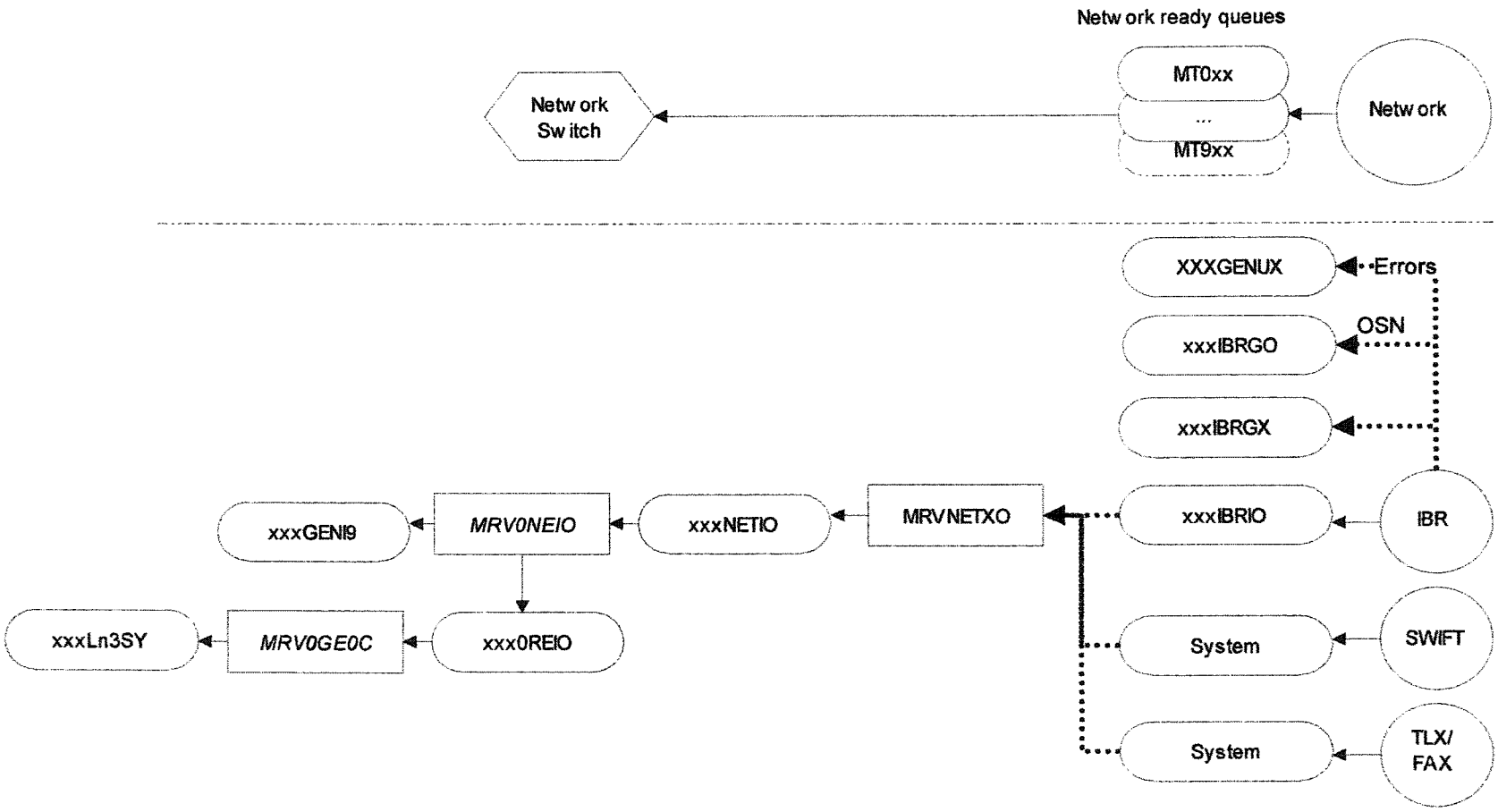


Figure 40 Output from Network (Fragment 1).

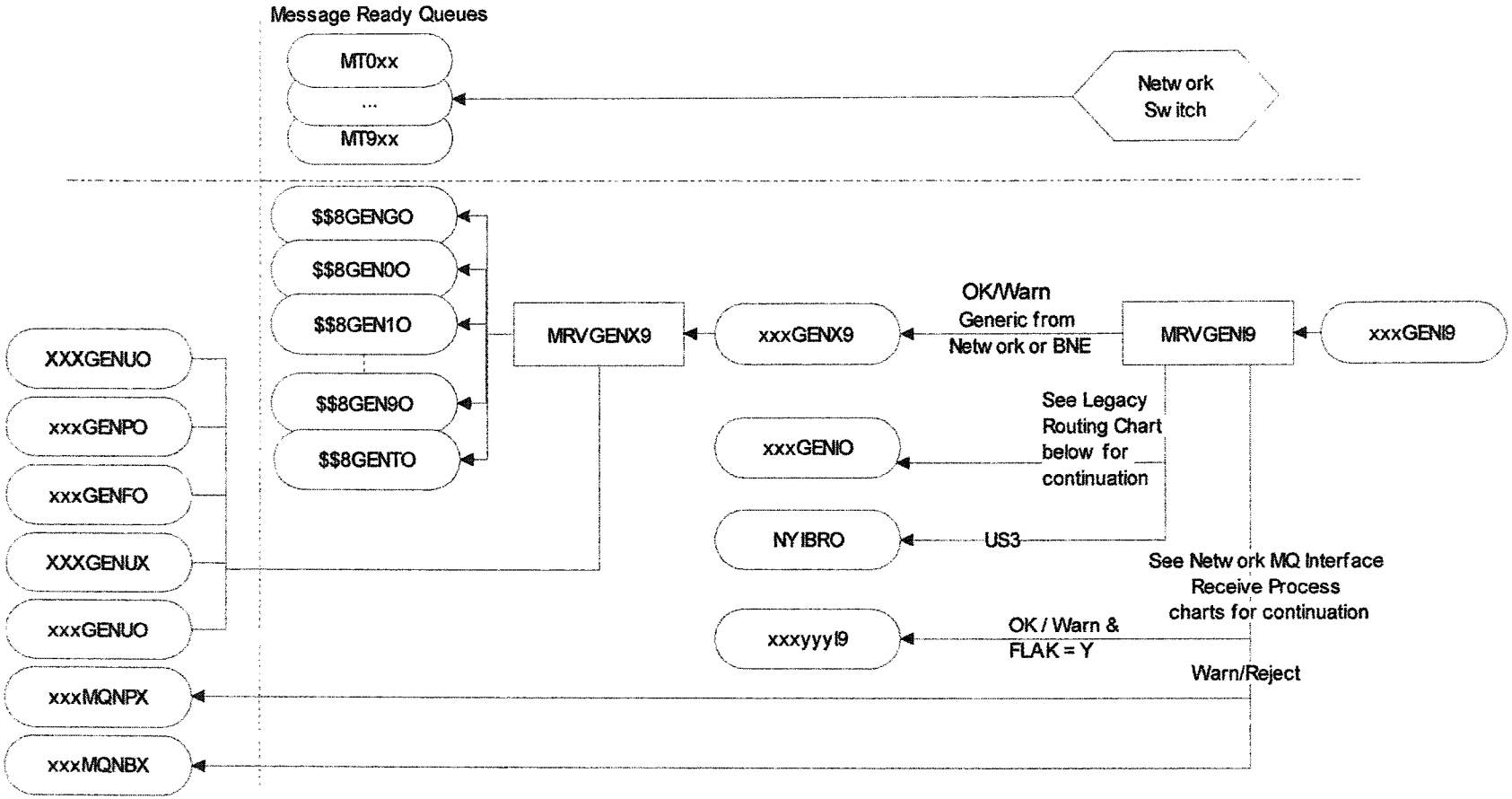


Figure 41 Output from Network (Fragment 2).

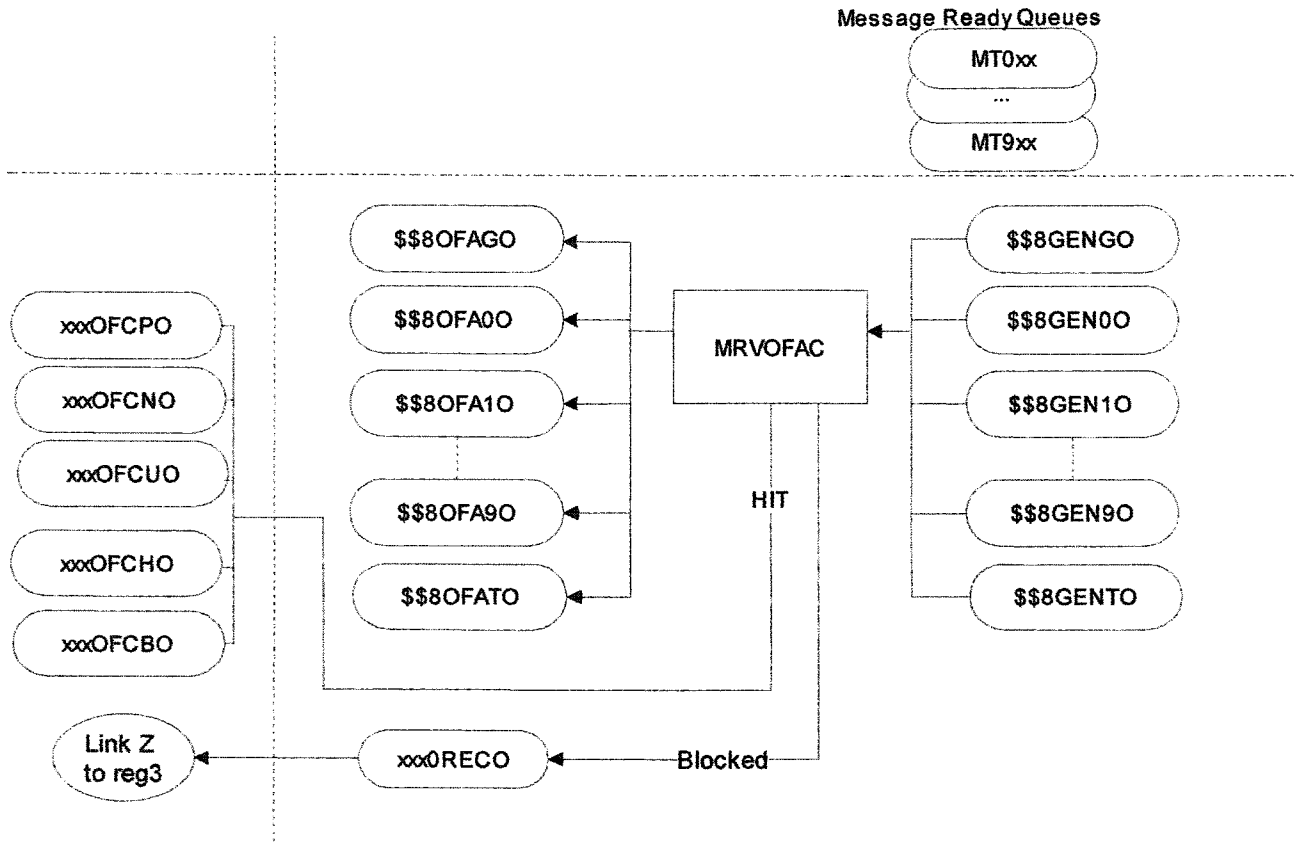


Figure 42 Output from Network (Fragment 3).

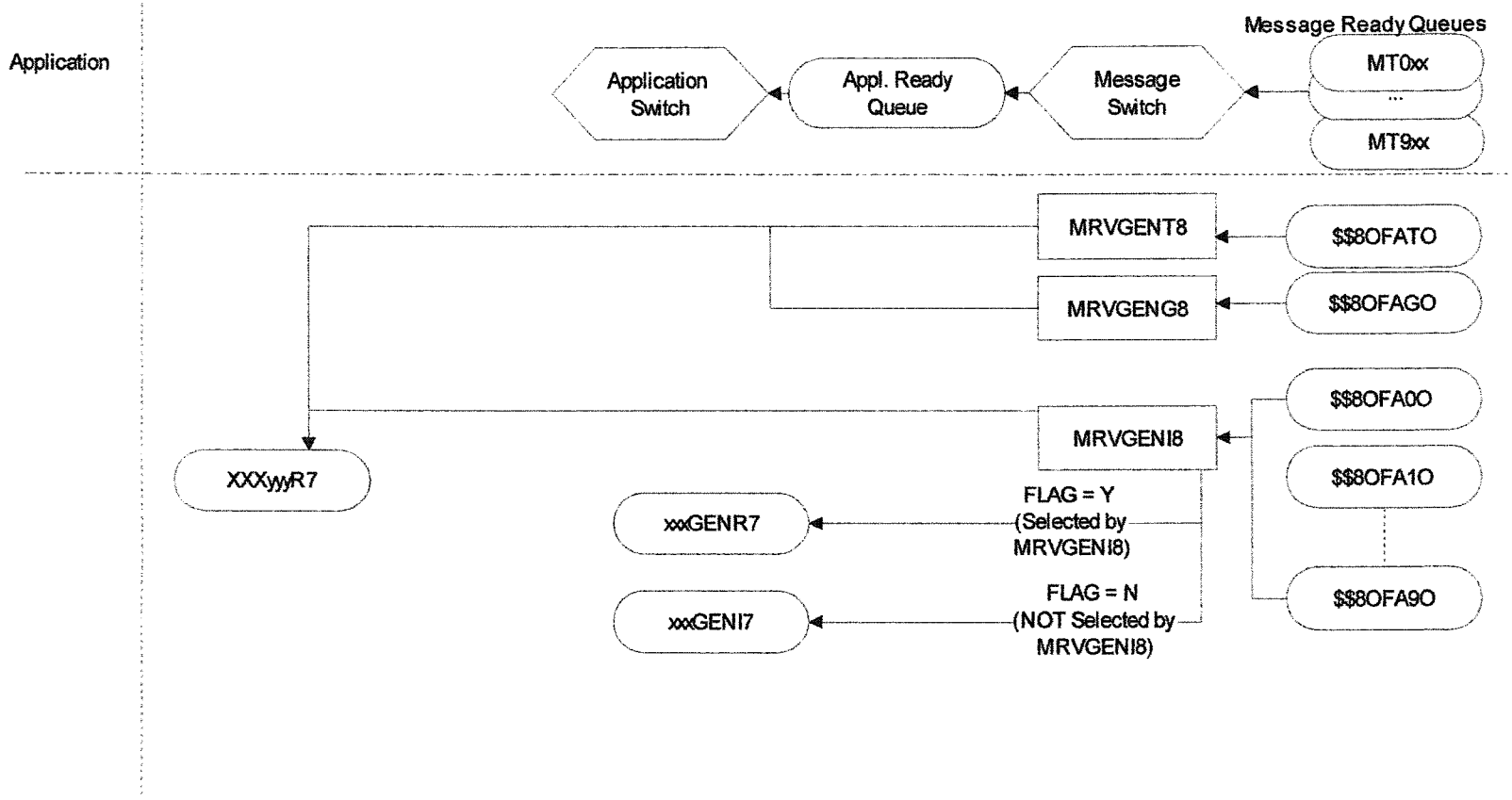


Figure 43 Output from Network (Fragment 4).

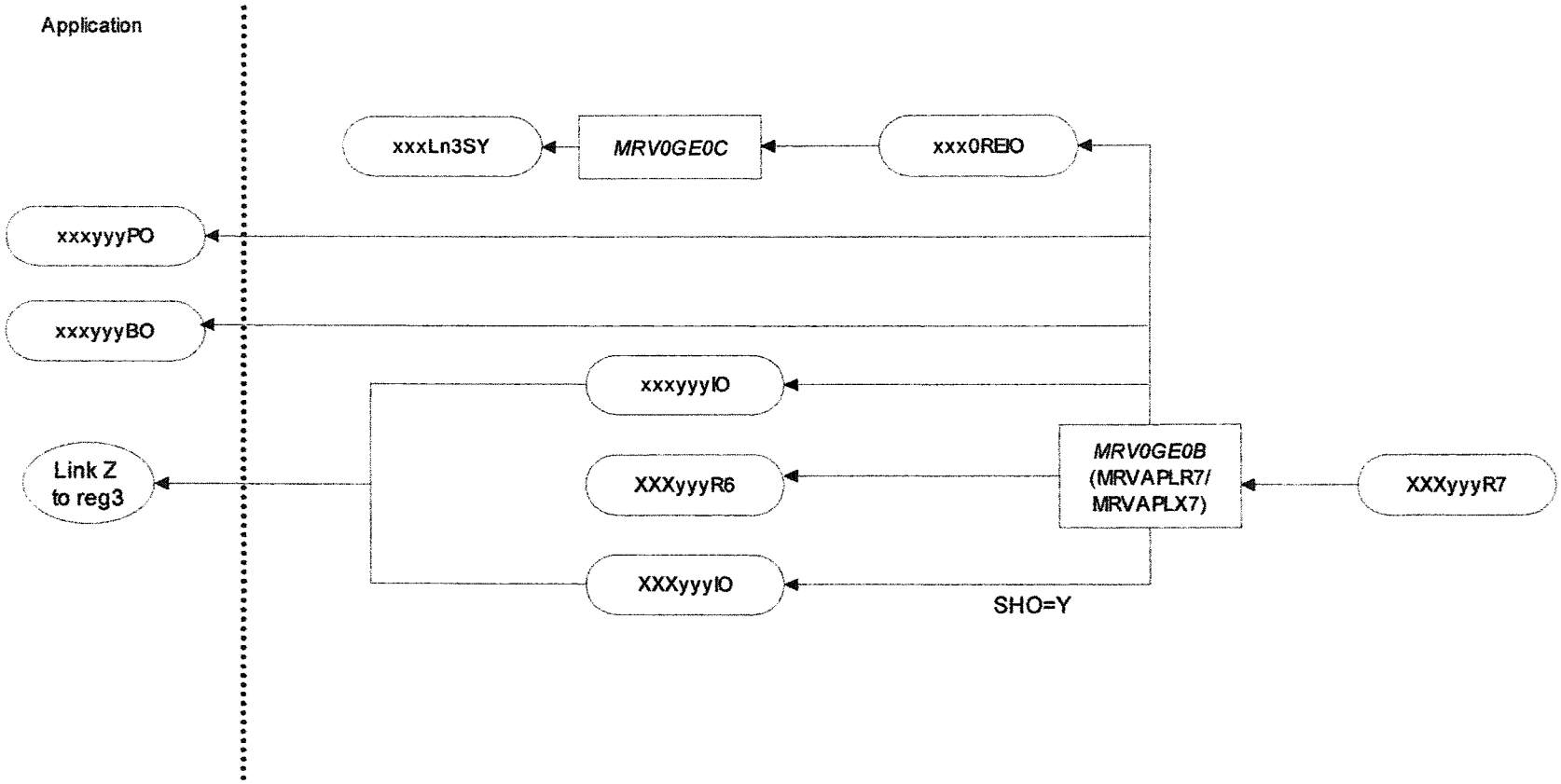


Figure 44 Output from Network (Fragment 5).

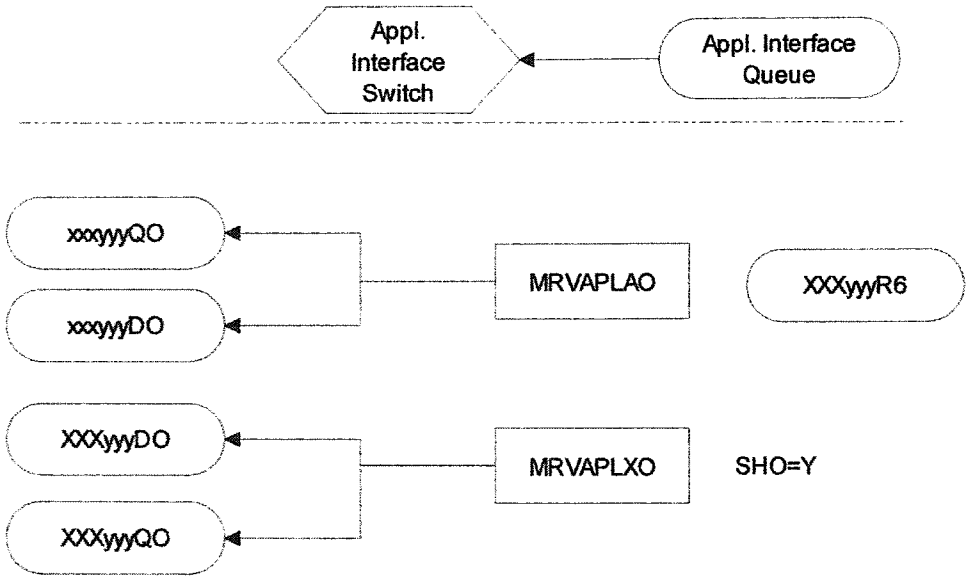


Figure 45 Output from Network (Fragment 6).

2.3.2 Application Interface Switch

Application Interface Switch (AIS) is a set of interface protocols and underlying software to link back-end applications and BMS. Functions of AIS are the following:

- receive messages from a back-end application,
- store messages on proper application queues for further processing,
- send ACKs for processed messages to proper applications,
- send messages to proper back-end applications ,
- receive UACK from a back-end application and reconcile.

Considering a financial messages as undividable unit of work for data exchange there are two main cases:

File transfer

Message Transfer

The length of a message varies depending on a protocol. Normally a message targeted for SWIFT does not exceed 10K bytes, a message for TELEX is not more than 32K bytes. If message exceeds the maximum length, it can be broken into multipart message by back-end application or by BMS. Each part of a multipart message is considered as a separate

message for delivery (for ex.. ½, 2/2). It can be reassembled into one message by a recipient. Transformation engine can also assemble/disassemble long messages.

A file can be stored on any secondary storage device: DASD pack, optical disk, magnetic tape, CD, DVD, floppy disk, hard drive, etc. It could spread over multiple volumes on the same device. Thus the size of a file is limited by the size of a storage used and can be measured by M-bytes or G-bytes. The size of a file is determined by the following:

- business needs for each batch transmission of messages at certain time of a day,
- Underlying FTP protocol for uninterrupted transmission.

FTP is being done from any to any computer platform (platform independent) and becomes the fastest and cheapest transferring mechanism for large batch of messages (scalable). Examples of FTP software packages: NDM (Network Data Mover by Computer Associates), Telnet, ZMODEM, Kermit. SWIFT STN offers IFT (Inerbank File Transfer) and SWIFTNet FileAct.

Disadvantages of FTP: each message has to wait for transmission until the whole batch of messages is completed.

AIS uses the simple sequencing protocol to check the proper file transfer for each application. Each file contains header ant trailer information.

Message transfer (as oppose to transfer of file of messages) is instantaneous, i.e. a message is being transferred as soon as it is completed and

released. The speed is limited by underlying network protocol. For example, MQ is an IBM product from any to any platform using TCP/IP. Communications between CICS mainframe regions use LU 6.2.

During file or message transfer some of the presentation layer functions are performed at AIS. To achieve platform independence computer-specific data formats are being converted from one platform to another if needed. For example, from ASCII on PC/VAX to EBCDIC on IBM mainframe.

To accommodate conversion between application-specific (external) and BMS-specific (internal) format the Transformation Engine is being employed.

Priority mechanism can be used to differentiate between application interfaces. Input/output queues on both sides of AIS can be prioritized. For example, xxxGPSQI and XXXGPSQO queues can have higher priority than xxxGSPQI and XXXGSPQO.

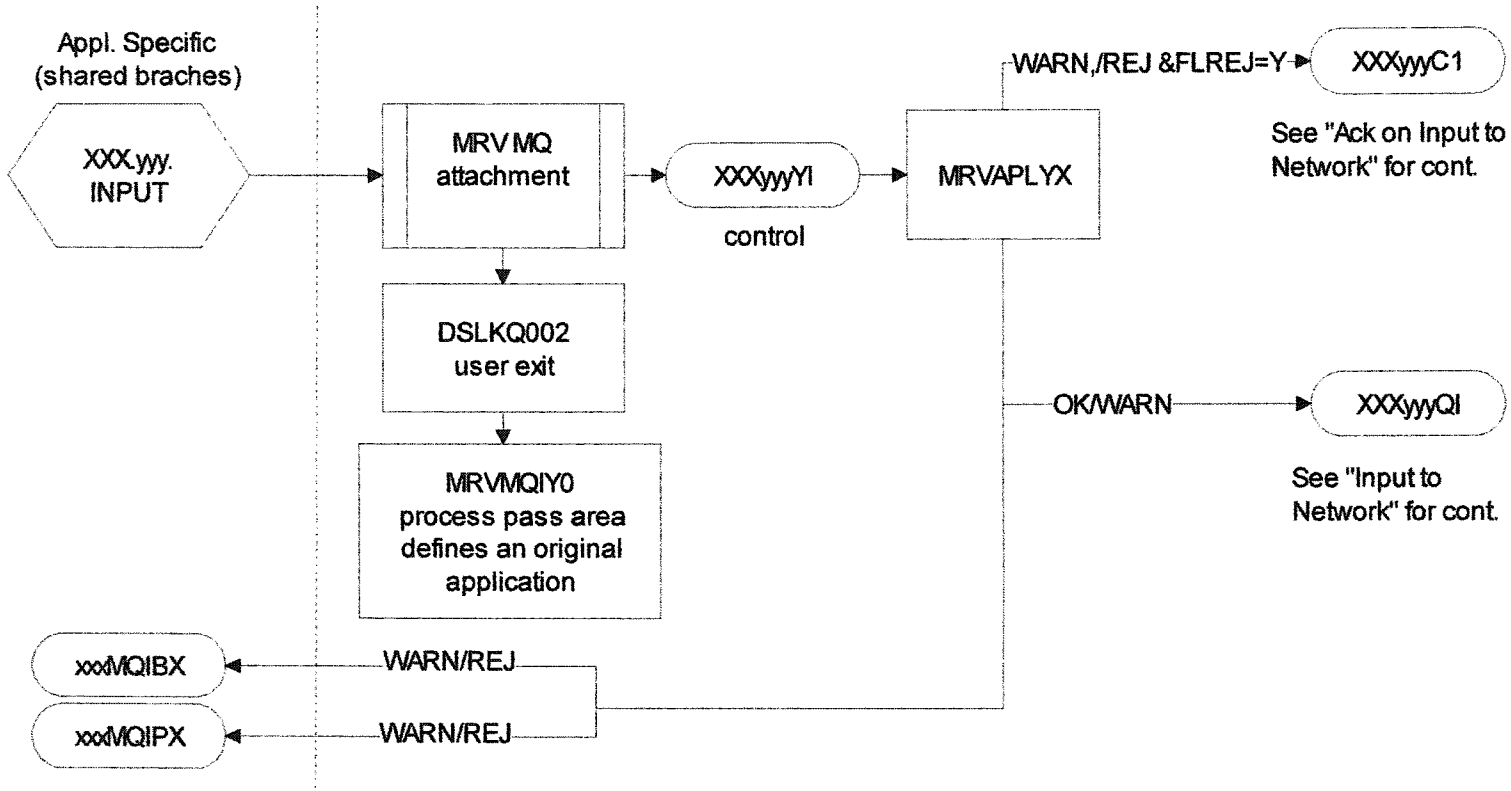


Figure 46 Application MQ Interface Receive Process - Input to Network.

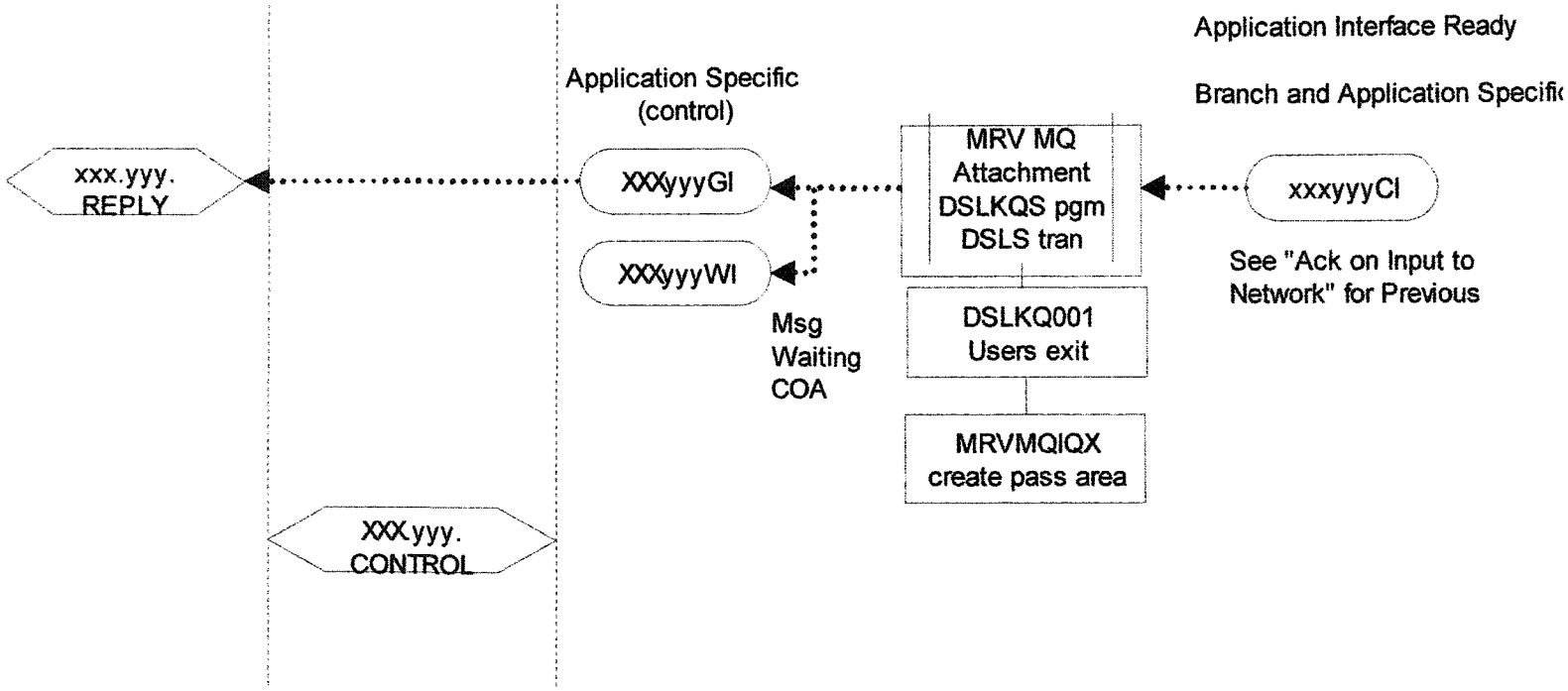


Figure 47 Application MQ Interface Acknowledgement (Rejection) Process - Input to Network.

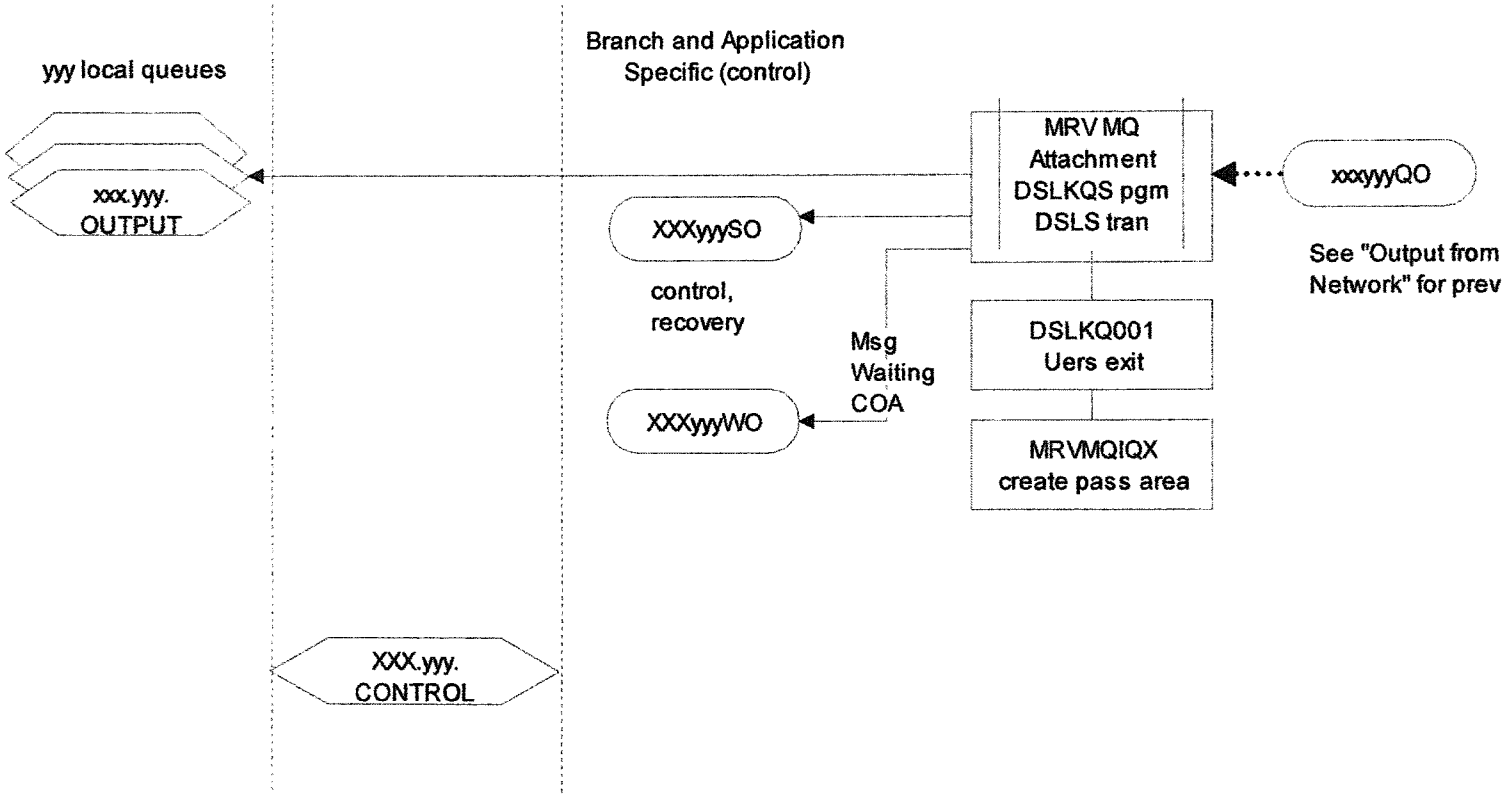


Figure 48 Application MQ Interface Send Process - Output from Network.

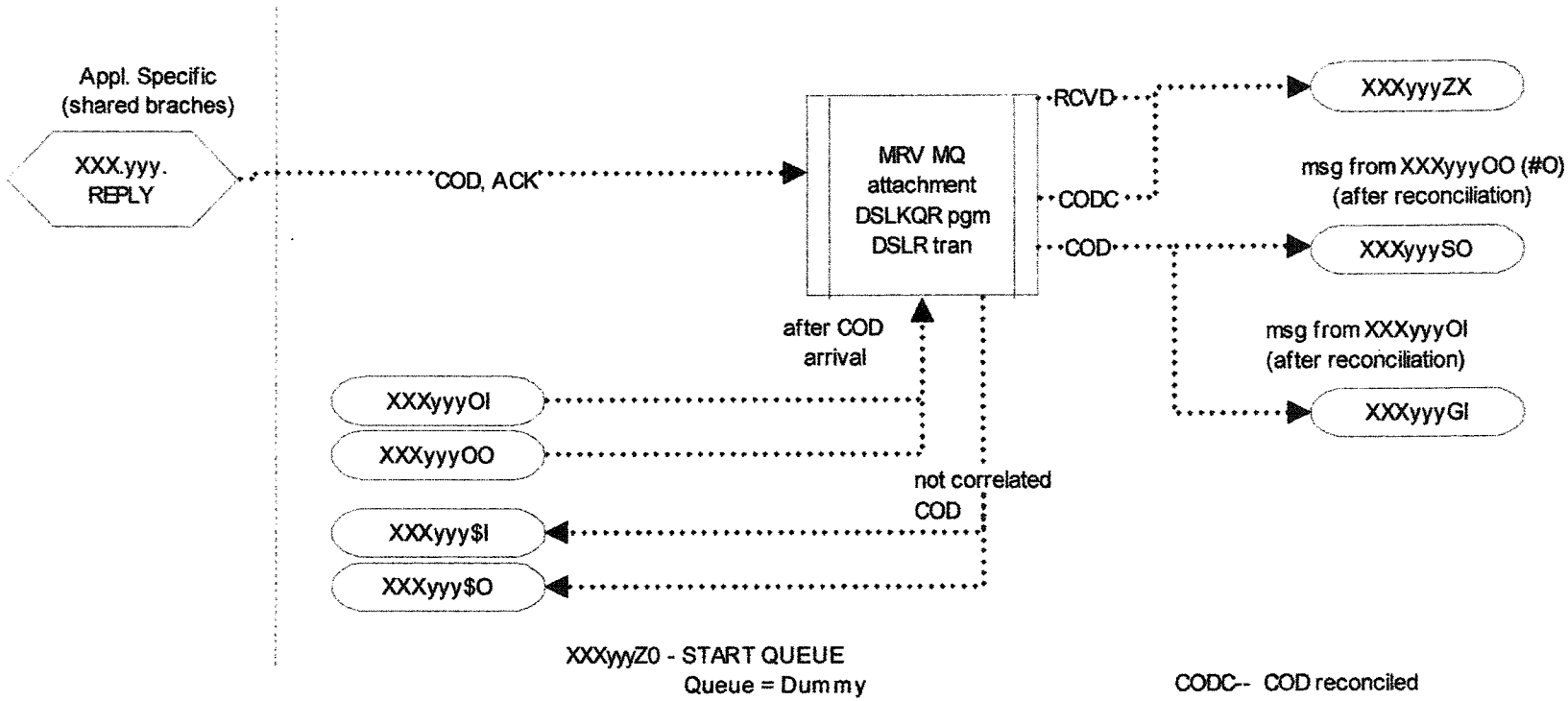


Figure 49 Application MQ Interface Reply Process – Network Input/Output (Fragment 1).

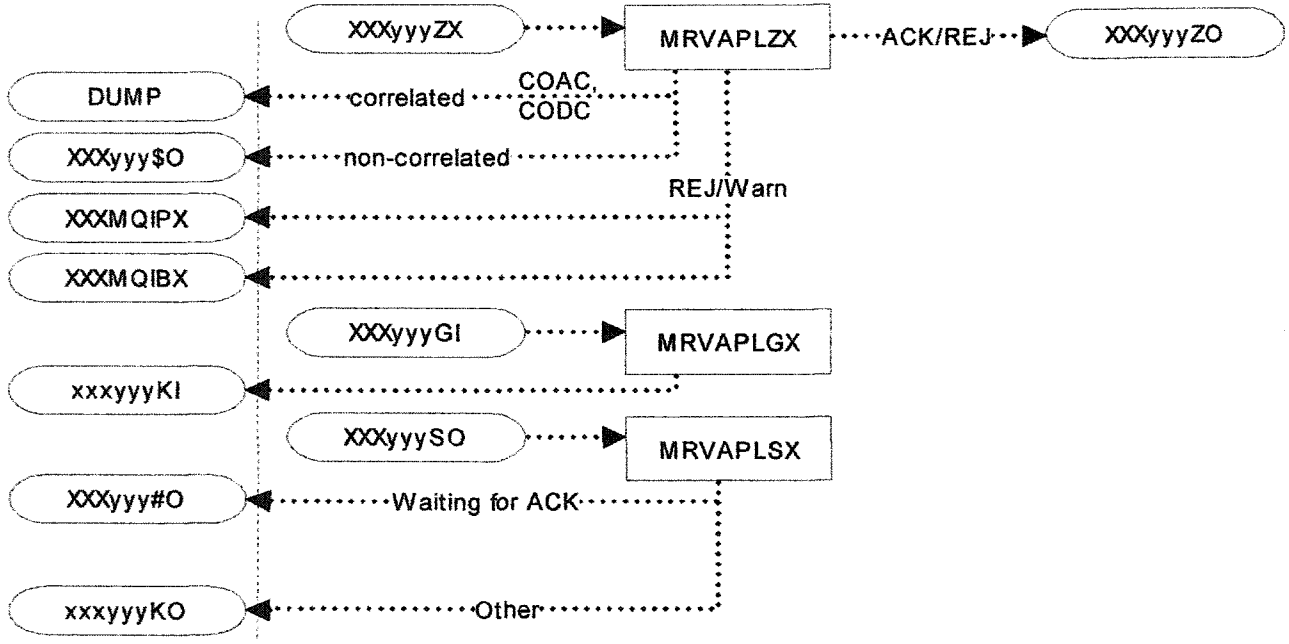


Figure 50 Application MQ Interface Reply Process – Network Input/Output (Fragment 2).

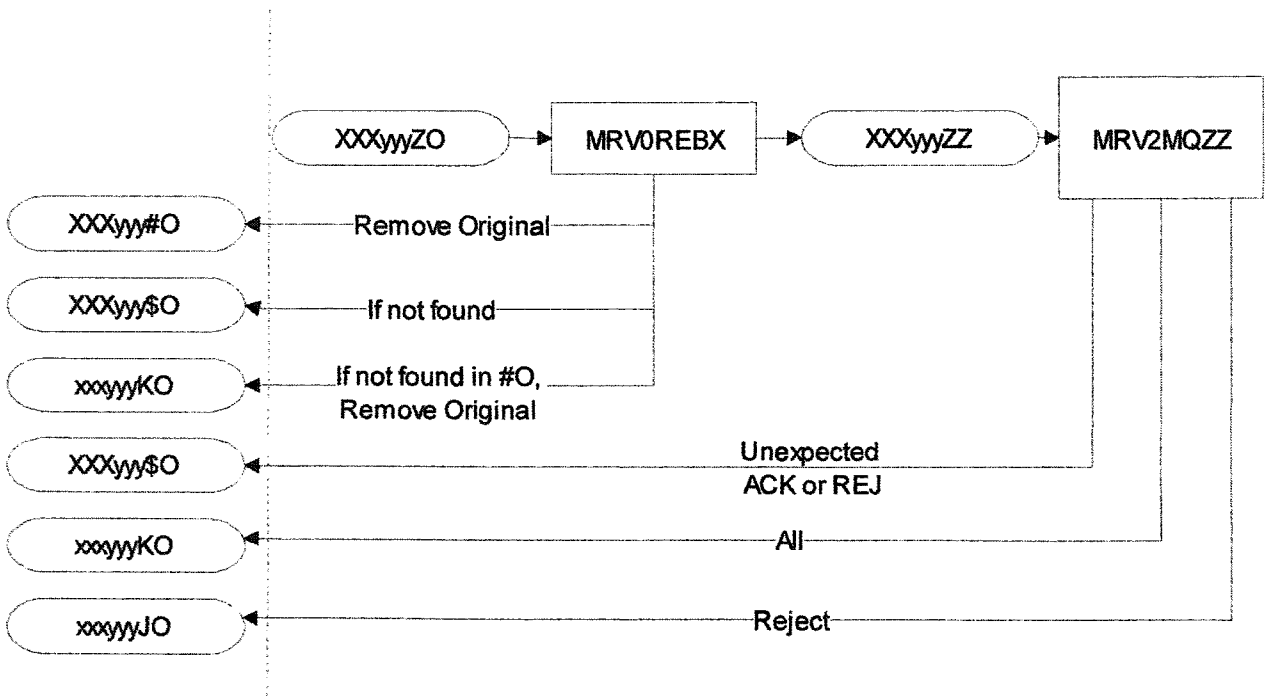


Figure 51 Application MQ Interface Reply Process – Network Input/Output (Fragment 3).

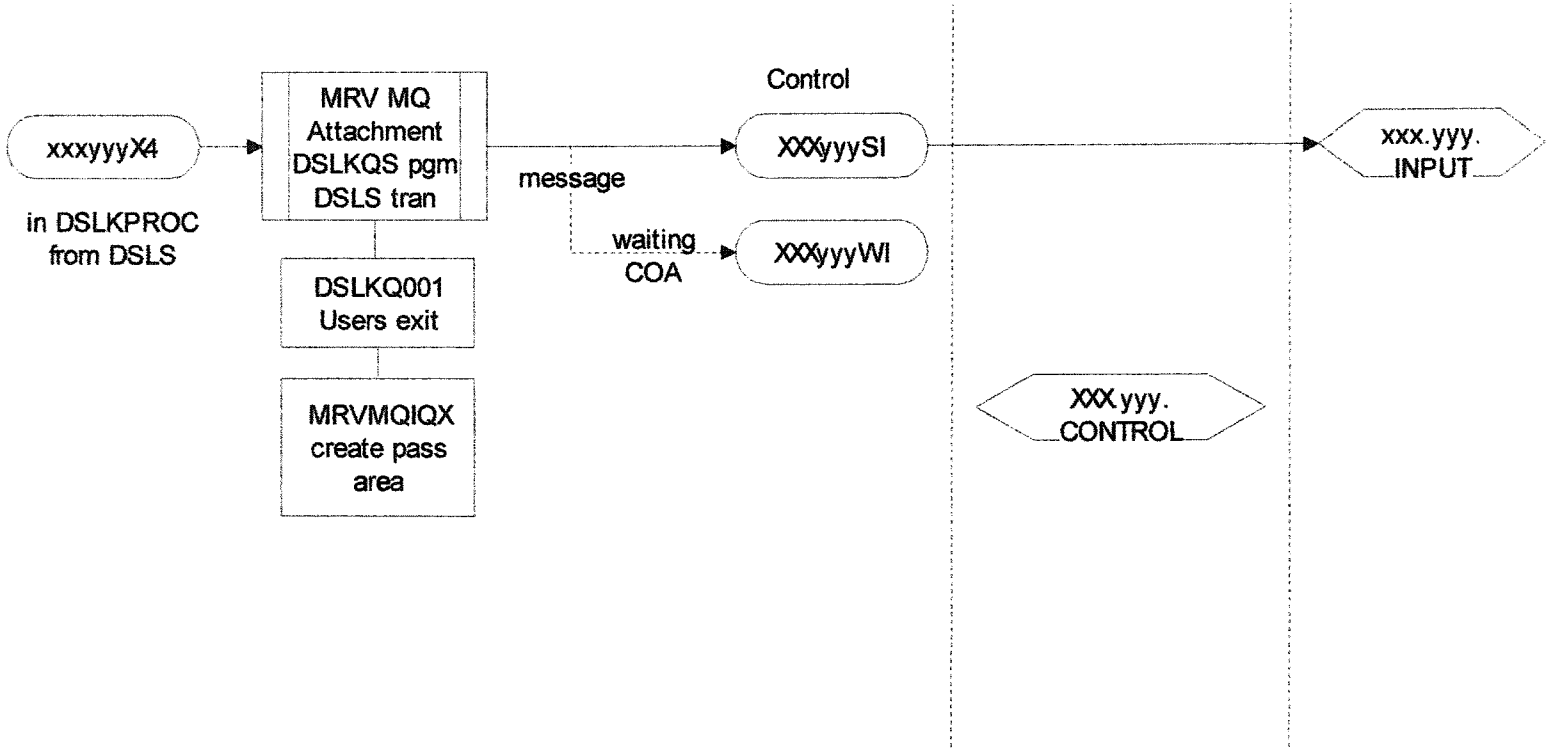


Figure 52 Network MQ Interface Send Process – Input to Network (Fragment 1).

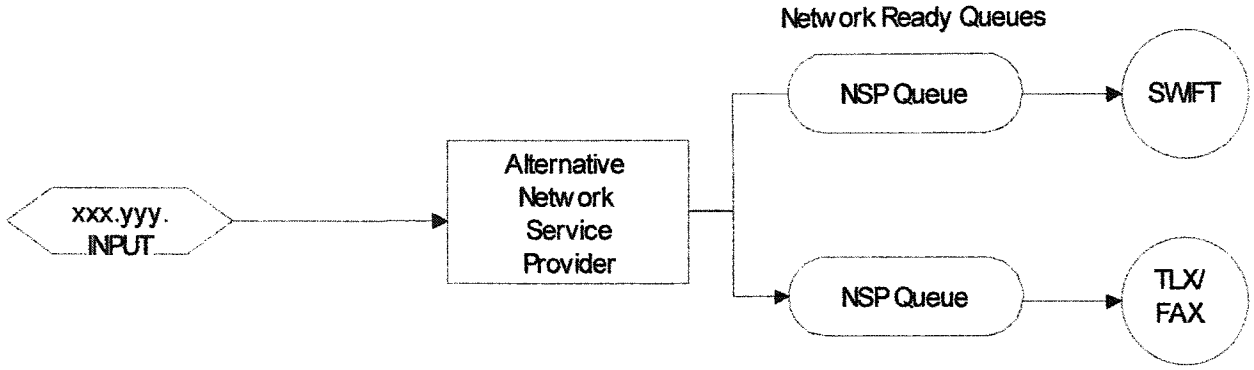


Figure 53 Network MQ Interface Send Process – Input to Network (Fragment 2).

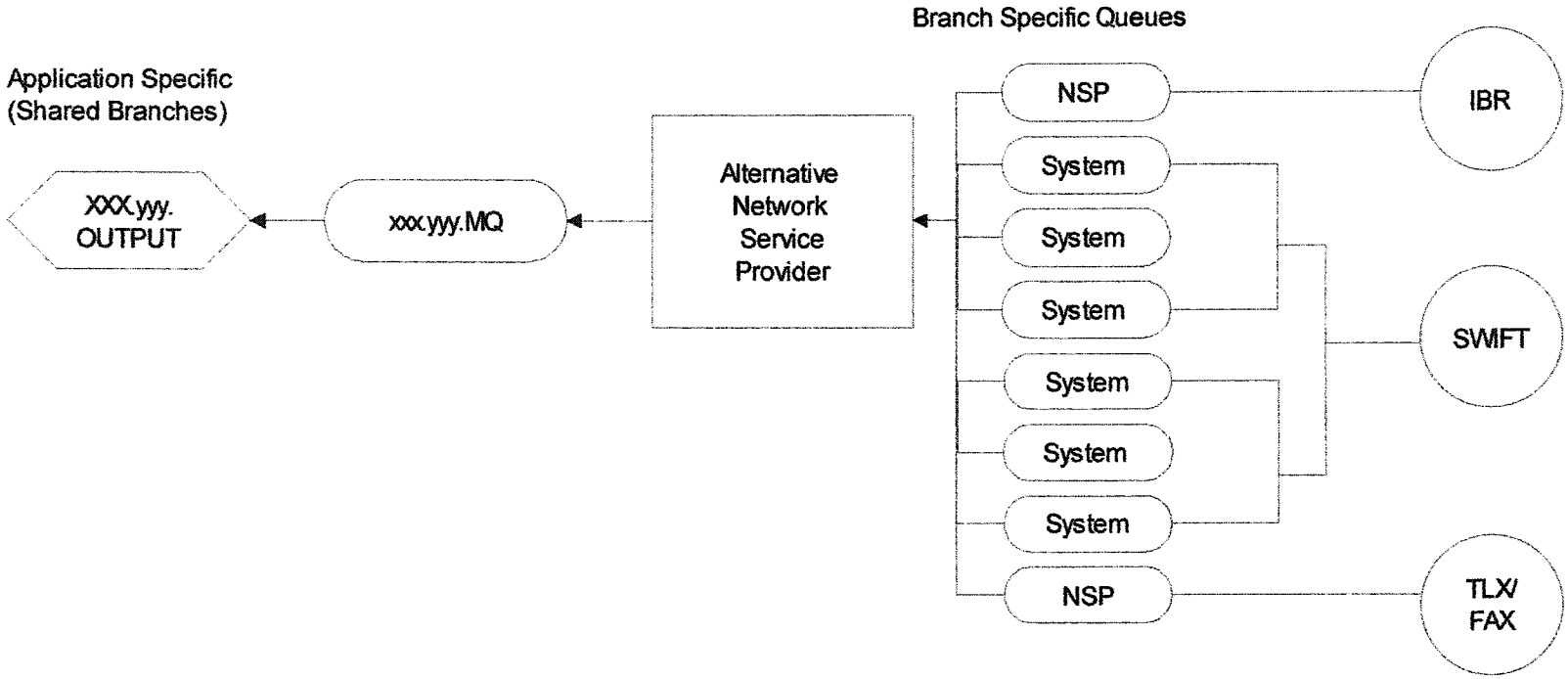


Figure 54 Network MQ Interface Receive Process – Output from Network (Fragment 1).

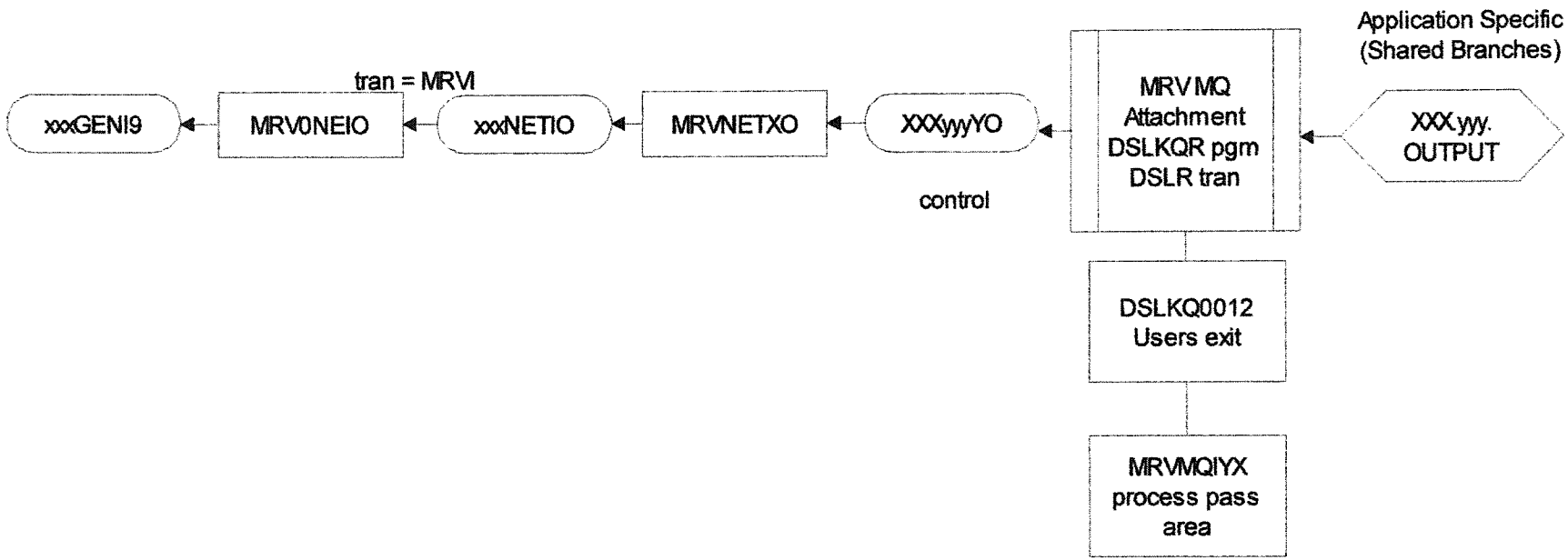


Figure 55 Network MQ Interface Receive Process – Output from Network (Fragment 2).

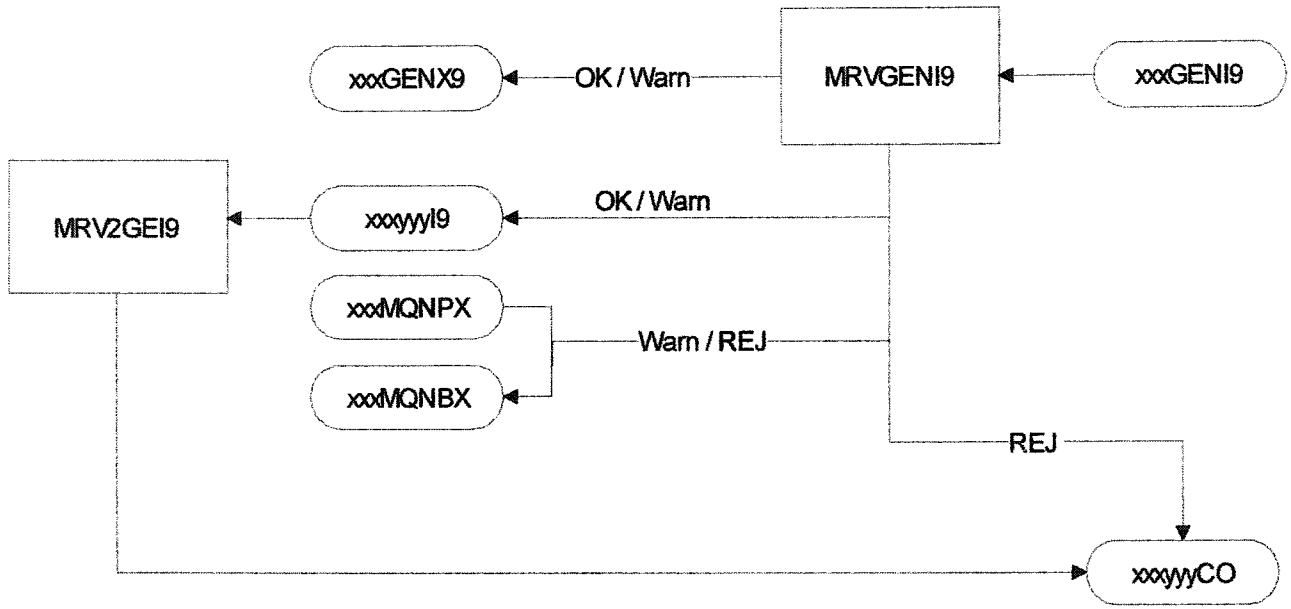


Figure 56 Network MQ Interface Receive Process – Output from Network (Fragment 3).

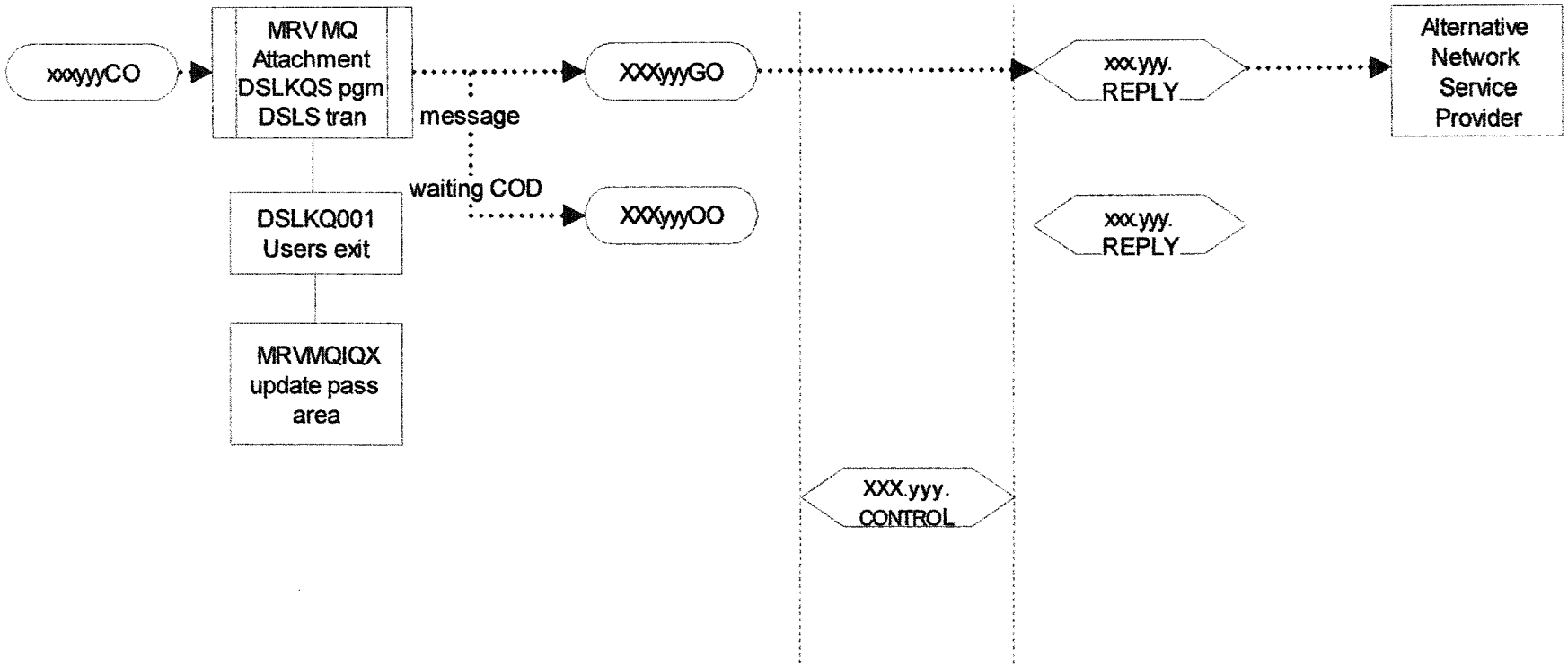


Figure 57 Network MQ Interface Rejection Process – Output from Network (Fragment 4).

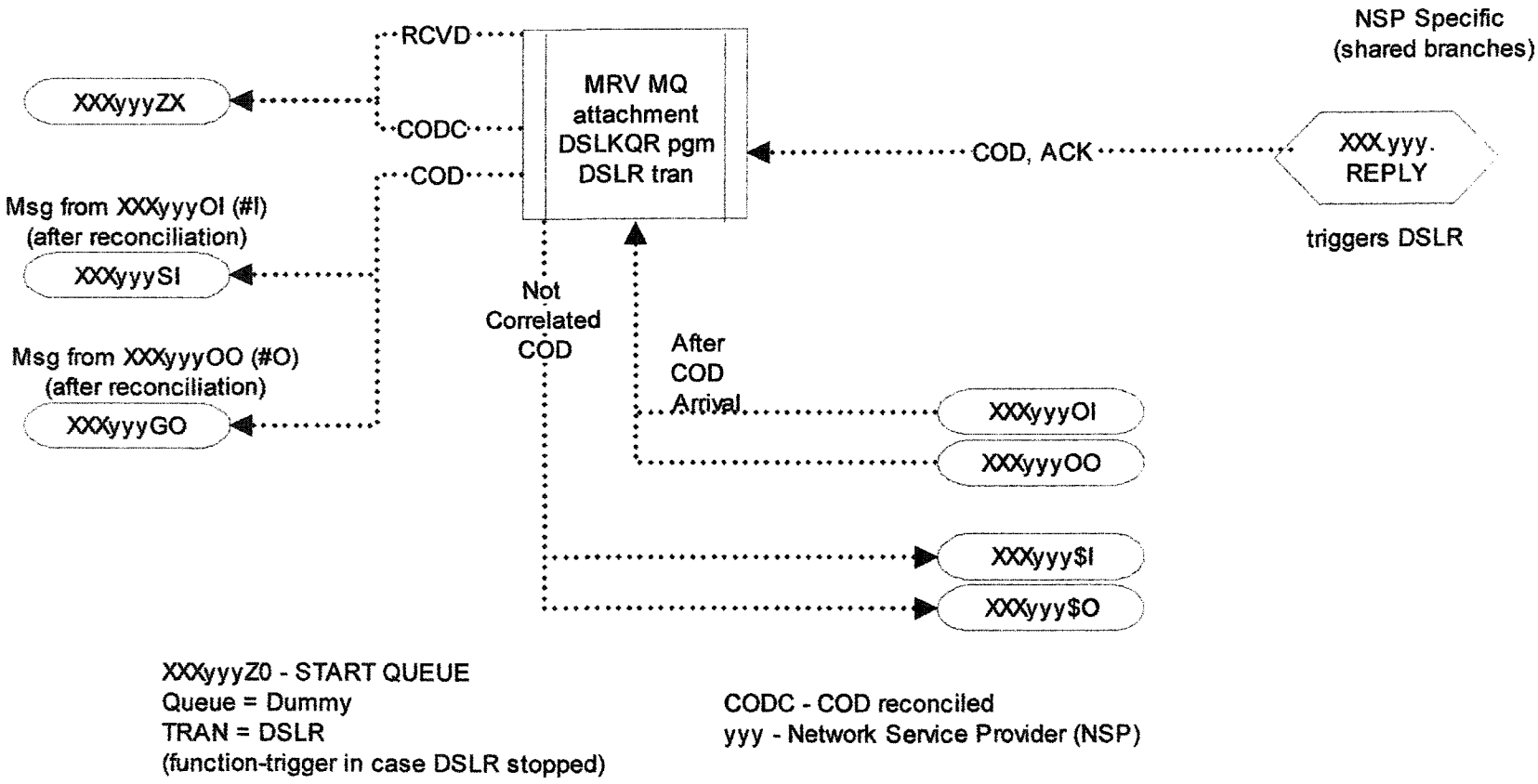


Figure 58 Network MQ Interface Reply Process – Network Input/Output (Fragment 1).

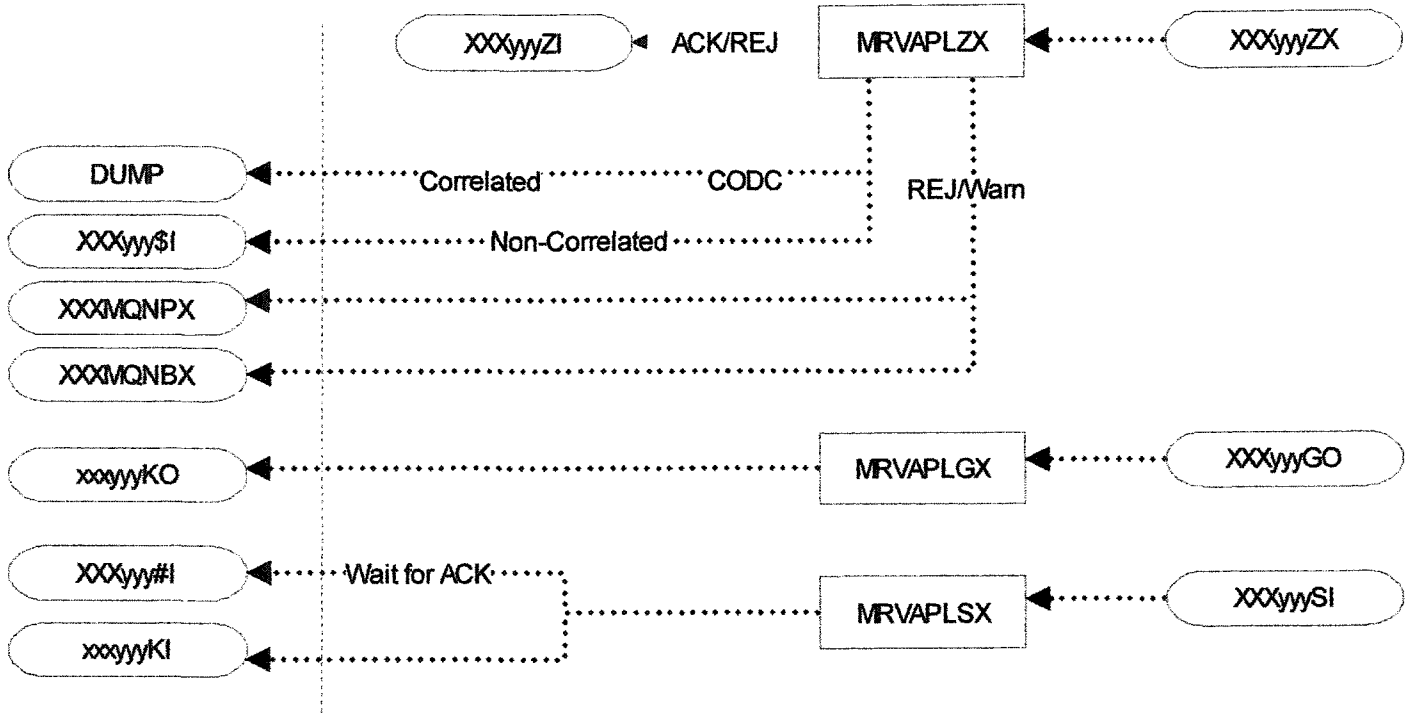


Figure 59 Network MQ Interface Reply Process – Network Input/Output (Fragment 2).

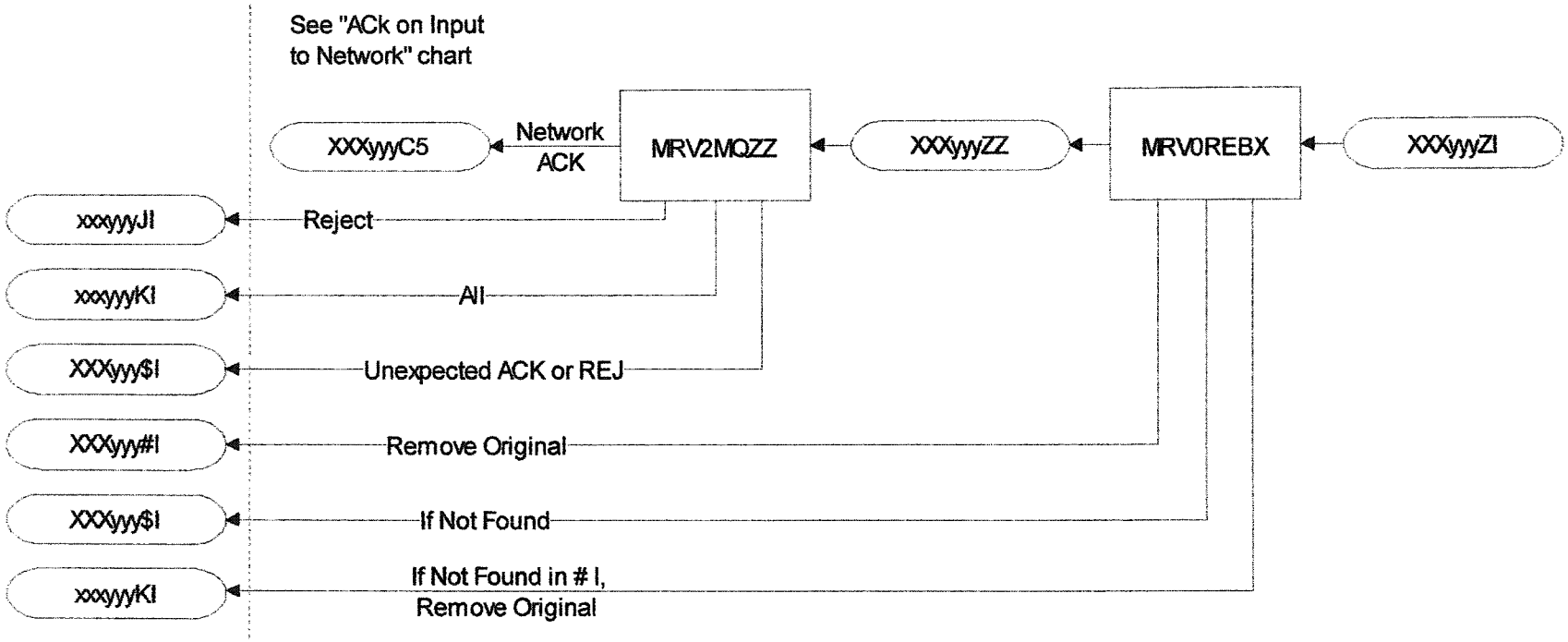


Figure 60 Network MQ Interface Reply Process – Network Input/Output (Fragment 3).

Protocol for MQ names for Bank ABC (running BMS on mainframe platform) communicating with Bank DEF is presented in Table 12. Example for MQ interface with GPS application is in Table 13.

Full Name:
[From MQ Manager Bank 1] [To MQ Manager Bank 2] [To Application] [From Application].[Branch Code].[Direction]

Alias Name:
<BMS>.<From Application>.<Branch Code>.<Direction>

No	MQ Process Type	MQ Queue Full Name	MQ Queue Alias Name
1	SEND Process (Output from Network to Application)	MQMABC . MQtDEF . yyy . BMS . xxx .OUTPUT	BMS.yyy.xxx.OUTPUT
2	ACKNOWLEDGEMENT (ACK/NAK on Input to Network from Application	MQMABC . MQtDEF . yyy . BMS . xxx . REPLY	BMS.yyy.xxx.REPLY
3	RECEIVE Process (Input to Network from Application	MQfDEF . MQMABC .BMS . yyy . ///. INPUT	BMS.yyy.///.INPUT
4	REPLY Process (Reply from Application to Network)	MQfDEF . MQMABC .BMS . yyy . ///. REPLY	BMS.yyy.///.REPLY
5	CONTROL queue format in SEND Process	MQMABC . MQMABC .BMS . BMS. yyy . CONTROL	BMS.BMS.yyy.CONTROL

Where: xxx – branch code (/// - for shared branches),
yyy – application code,
t, f – to/from platform: V – for VAX , M – for IBM Mainframe,
s – for UNIX (assuming Bank ABC runs BMS on Mainframe).

Table 12 MQ Names Format.

No	MQ Process Type	MQ Queue Full Name	MQ Queue Alias Name
1	SEND Process (Output from Network to Application)	MQMABC.MQVABC.GPS.BMS. BEB.OUTPUT	BMS.GPS.BEB.OUTPUT
		MQMABC .MQVABC.GPS.BMS. DEF.OUTPUT	BMS.GPS.DEF.OUTPUT
		MQMABC .MQVABC.GPS.BMS. GB2.OUTPUT	BMS.GPS.GB2.OUTPUT
		MQMABC .MQVABC.GPS.BMS. LUL.OUTPUT	BMS.GPS.LUL.OUTPUT
		MQMABC . MQVABC. GPS. BMS. JPJ.OUTPUT	BMS.GPS.JPJ.OUTPUT
2	ACKNOWLEDGMENT (ACK/NAK on Input to Network from Application)	MQMABC . MQVABC. GPS. BMS. BEB.REPLY	BMS.GPS.BEB.REPLY
		MQMABC . MQVABC. GPS. BMS. DEF.REPLY	BMS.GPS.DEF.REPLY
		MQMABC . MQVABC. GPS. BMS. GB2.REPLY	BMS.GPS.GB2.REPLY
		MQMABC . MQVABC. GPS. BMS. LUL REPLY	BMS.GPS.LUL.REPLY
		MQMABC . MQVABC. GPS. BMS. JPJ.REPLY	BMS.GPS.JPJ.REPLY
3	RECEIVE Process (Input to Network from Application)	MQVABC . MQMABC .BMS . GPS . ///. INPUT	BMS.GPS.///.INPUT
4	REPLY Process (Reply from Application to Network)	MQVABC . MQMABC .BMS . GPS . ///. REPLY	BMS.GPS.///.REPLY
5	CONTROL queue format in SEND Process	MQMABC.MQMABC.BMSBMS.GPS. CONTROL	BMS.BMS.GPS.CONTROL

Note: GPS runs on VAX platform for Bank ABC.

Table 13 MQ GPS Names Example.

2.3.3 Application Ready Switch

Application Ready Switch (ARS) implements business rules for each application. Functions of ARS are the following:

- Receive a message from application interface ready queue XXXyyyR1 on the way to network (input to network).
- Process the received message according to the business rules (input to or output from network).
- Store processed message on application ready queue XXXyyyR2 on the way to network.
- Receive network ACK from application ready queue XXXyyyC2.
- Reconcile received ACK with original message and store it on application interface ready queue XXXyyyC1.
- Receive a message from application ready queue XXXyyyR7 on the way from network (output from network).
- Store processed message on application interface ready queue XXXyyyR6 on the way from network.

The basic business rules for ARS are simple:

- backup messages on xxxyyyBI or xxxyyyBO depending on the direction of a message for subsequent investigation, resend, archival. Normally ACKs are stored on xxxyyyBI and UACKs on xxxyyyBO after reconciliatioin;

- print messages. Typically ACKs and UACKs are printed.

Some exceptions apply. For example, for Confirmation Matching System (CMS) messages are removed from further processing on input to network and delivered to the CMS application instead. Messages for payment confirmations will arrive from network later and will be delivered to CMS for matching with originals.

Priority mechanism can be used to differentiate between applications. Input/output queues on both sides of ARS can be prioritized. For example, XXXGPSR1 and XXXGPSR7 queues can have higher priority than XXXGSPR1 and XXXGSPR7.

2.3.4 Message Ready Switch

Message Ready Switch (MRS) on input to network departs from application concept and comes to network unit of work which in most cases is a message. For output from network MRS just does the reverse.

Functions of MRS are the following:

- Receive a message from application ready queue XXXyyyR2 on the way to network (input to network).
- Process the received message applying value-added services (such as OFAC scanning) if needed (input to or output from network).
- Store processed message on message ready queues according to the message type ($$3OFAnI) on the way to network.

- Receive network ACK from message ready queue xxxGENC4.
- Reconcile received ACK with original message and store it on message ready queue xxxGENC3.
- Receive a message from message ready queue xxxGENI9 on the way from network (output from network).
- Identify an application yyy for a message applying business rules.
- Store processed message on application ready queue XXXyyyR7 on the way from network.
- Archive messages for each location on xxxGENLX (input/output).

The simple business rules for MRS are tables of message types belonging to each application. They can also include some conditions in basic headers or in content of messages. There can be one to many relationship, i.e. one message can belong to more than one application on output from network. The reverse is not true. Any particular message on the way to network is generated by one and only one application.

Priority mechanism can be used to differentiate between message types. Input/output queues on both sides of MRS can be prioritized. For example, $$3GEN2I and $$3GEN2O queues can have higher priority than $$3GEN5I and $$3GEN5O.

2.3.5 Network Selection Switch

Network Selection Switch (NSS) makes the choice of network transparent to the end users. NSS on input to network implements the best method in selecting the network. For output from network NSS combines traffic from different networks for each connected location and passes that traffic for further processing by MRS.

Functions of NSS for input to network are the following:

- Receive a message from message ready queue $$3OFCnI.
- Verify the availability of the networks.
- Select a network and network service applications based on some business criteria ('best method").
- Select a line within a network for multiple lines connection.
- Store a message on the network ready queue according to the selected network service application, transmission line, message priority and connected location.
- Receive a network ACK and store it on the combined queue xxxGENC4 per location (to be processed by MRS).

The business criteria for routing decision can be the following:

- type of service subscribed by sender and recipient (SWIFT, Telex, FAX), cable, regular mail);

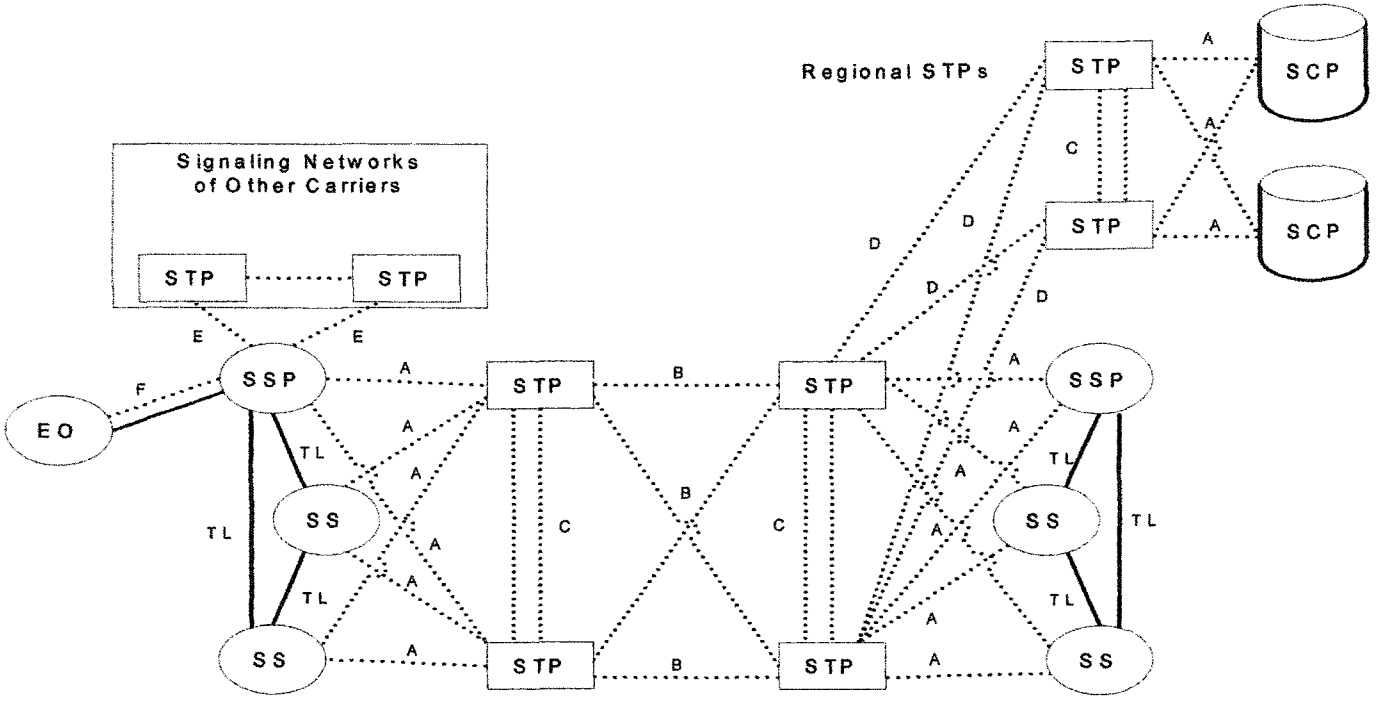
- availability of a carrier;
- congestion on a line (switch to another line if available),
- source and destination of information;
- message format (SWIFT, XML, etc.)
- minimal cost;
- minimal delay;
- reliability.

To accommodate conversion between BMS-specific (internal) format and network-specific (transmission) format the Transformation Engine can be employed. But in some cases there no formal transformation rules (maps) allowing to generate, for instance, SWIFT-specific ISO 15022 format. In that cases a message is sent TELEX if TELEX is defined the second best after SWIFT. In other cases the message indicator might negate the "best method" and instruct the BMS on how to deliver the message, for ex. by FAX.

Priority mechanism can be used to differentiate between messages on input to network. For example, payments have higher priority than other messages, security settlements have higher priority than other security messages. NSS store messages on network ready queues according to their priority. Network ready queues are prioritized for each transmission line.

2.4 Message Transfer Point (MTP)

The MTP performs different functions depending on a communication layer it operates. For lower layers (network, data link, physical) it interconnects similar to STPs (see Figure 61).



SCP = ServiceControl Point; STP = Service Transfer Point; SSP = Service Switching Point; SS = Switching System; EO = End Office; TL = Trunk Lines; A = Access Link; B = Bridge Link; C = Cross Link; D = Diagonal Link; E = Extended Link; F = Fully-Associated Link.

Figure 61 Interconnection of STPs in the CCS7 Signaling Network.

The MTP transfers queries and responses between MSPs and MCPs of one or multiple BMS instances facilitating message progress through the system. The transfer takes place via MCS links to dispatch the context-dependent query to the right MCP for quick response (correspondence file, currency information, authenticator key, etc.). The functions of MTP are normally controlled by BMS residential software modules.

2.4.1 Functions at Each Switching Layer

The MTP performs layer-specific functions for each of four sub-layers of FIIN application layer: Examples of basic queries are the following:

1. Application Interface:

   -validity and availability of a Server for batch file transfer,

   -validity and availability of MQ Manager for online message interaction,

   -existence of sessions and queue names,

   -access MCP for expected/received sequence number, message persistency, commit level, max queue depth.

2. Application Processing:

   -existence of location/application combination,

   -business rules.

3. Message Processing:

   -message standards data base,

   -message type-specific data base (OFAC).

4. Network Selection:

   -availability of a network,

   -availability and throughput of active lines.

2.4.2 Links Between Message Transfer Point and Carrier's Service Transfer Point at Network Selection Switch

The LOGIN command is sent to the selected Network. FIIN MTP provides the response to fill this command with the information from MCP specific to the sender: LSN (LOGIN session number), MAC trailer, CHK, etc.. Network accept the LOGIN and its STP verifies it against SCP.

The details of communications between Computer-Based Terminal (CBT) and SWIFT's FIN are shown in the Table 14.

Abbreviation	Origin	Name	BMS Flows
LOGIN	CBT	Login message (APDU 02)	Operator issues command Login via End User Driver. BMS creates message L02: 1. 501 user-authentication: text used to authenticate the user connecting to SWIFT (MAC trailer is calculated based on SLS keys). 2. 110 window-size: Maximum number of messages sent or received without waiting for or sending acknowledgement. 3. 330 current-session-info {1:L02ABCDUS30AXXX}{4:{501:2712013801400000271201380140000FB93243F0000000000000000000000}{110:010}}
LOGIN -ACK	FIN	Login positive acknowledgement message (APDU 22)	BMS receives from SWIFT Login ACK message which contains: 1. 502 system-authentication MAC 2. 151 session-number Session Number allocated to the new GPA session. 3. 110 window-size 4. 333 previous-session-info Previous session information, including: • date last session opened • time last session opened • session number • date last session closed • time last session closed • reason for closure • last ISN received • last OSN sent BMS checks MAC trailer and updates internal BMS tables. LT GPA application is in the LOGIN status. {1:L22ABCDUS30AXXX}{4:{502:2717421001410142014101 41DCB5D8926BB3A42100011}{151:1630}{177:0310271242}{110:001}{333:0310271201162903102712090000000001000000}}

Abbreviation	Origin	Name	BMS Flows
			Note: Table continues from the previous page
LOGIN FIN -NAK		Login negative acknowledgement message (APDU 42)	BMS receives from SWIFT Login ACK message which contains: {1:L42ABCDUS30AXXX}{4:{502:2719445102 00014201420200AFDCE0F63518200F0001000 001}{503:L33}{177:0310271444}}
SELEC T	CBT	Select message (APDU 03)	Similar to Login: {1:A03ABCDUS30AXXX1631000002}{4:{101 :F}{501:27151452014800002715145201480000 84530269000000000000000000000000}{110:01 0}{204:YY}{208:Y}{338:SWFURGSWFSYSS WF1XXSWF2XXSWF3XXSWF4XXSWF5XX SWF7XXSWFA99SWFB99SWFOTH}}{5:{TN G:}}
SELEC T-ACK	FIN	Select positive acknowledgement (APDU 23)	Similar to Login ACK: {1:A23ABCDUS30AXXX1631000002}{4:{101 :F}{502:27201454014801490148014884530269 5B5AED7B0001000001}{151:1721}{177:0310 271514}{110:010}{204:YY}{208:Y}{338:SWF URGSWFSYSSWF1XXSWF2XXSWF3XXSW F4XXSWF5XXSWF7XXSWFA99SWFB99SW FOTH}{333:031023214917200310251520A902 95820224611}}
SELEC T-NAK	FIN	Select negative acknowledgement (APDU 43)	

Abbreviation	Origin	Name	BMS Flows
			Note: Table continues from the previous page
QUIT	CBT	Quit message (APDU 05)	Operator issues command Quit via End User Driver. BMS creates message F05: 1. 173 day-time {1:F05ABCDUS30AXXX1721295874}
QUIT-ACK	FIN	Select positive acknowledgement message(APDU 25)	BMS receives from SWIFT Quit ACK message which contains: 1. session-info Session information, including: • session number • date session opened • time session opened • date session closed • time session closed • reason for closure • quantity of messages sent • quantity of messages received • first ISN • last ISN • first OSN • last OSN 2. 401 error-code for Logout/Quit error codes. BMS updates internal BMS tables. LT FIN application is in the CLOSED status. {1:F25ABCDUS30AXXX1721295874}{4:{331:17210310271514031027154900000005400185729582129587422461222646 8}{401:02}}
LOGOUT	CBT	Logout message (APDU 06)	Operator issues command Logout via End User Driver. BMS creates message L06: 1. 173 day-time {1:A06ABCDUS30AXXX1630000001}

Abbreviation	Origin	Name	BMS Flows
			Note: Table continues from the previous page
LOGOUT-ACK	FIN	positive acknowledgement message (APDU 26)	BMS receives from SWIFT Logout ACK Logout message which contains: 1. 331 session-info Session information, including: • session number • date session opened • time session opened • date session closed • time session closed • reason for closure • quantity of messages sent • quantity of messages received • first ISN • last ISN • first OSN • last OSN 2. 401 error-code for Logout/Quit error codes. BMS updates internal BMS tables. LT GPA application is in the LOGOUT status. {1:A26ABCDUS30AXXX1630000001}{4:{331:1630031027124203102714360000000010000000000100000100000010000000}{401:02}}

Abbreviation	Origin	Name	BMS Flows
			Note: Table continues from the previous page
APC-OSN	FIN	APC output message (APDU 01 – A)	BMS receives from SWIFT O-type GPA message. DWSDGPA program checks OSN, MAC trailer, and routes message using routing table from DWSLTT. If OSN is not equal to expected OSN, BMS aborts session. If message is not authenticated, message is marked with DWS error and routed according to DWSLTT {1:A01ABCDUS30AXXX1633000001}{2:O05 22227031027MFMFXXXX0XXX00001127030 310271727}{4:{336:LTDIRA00000A}{336:CU RNTA00000}{336:FUTURA00000}{336:SWFS YS00000A}{336:SWFURG00002A}{336:SWF 1XX00073A}{336:SWF2XX00023A}{336:SW F3XX00000A}{336:SWF4XX00000A}{336:S WF5XX00000A}{336:SWF7XX00000A}{336: SWFA9900000A}{336:SWFB9900001A}{336: SWFOTH00211A}}{5:{CHK:A94A383E293E} {SYS:1727031027ABCDUS30AXXX16330000 04}{TNG:}}
FIN-OSN	FIN	FIN output message (APDU 01 – F)	BMS receives from SWIFT O-type FIN message. DWSDGPA program checks OSN, MAC trailer, and routes message using routing table from DWSLTT. If OSN is not equal to expected OSN, BMS aborts session. If message is not authenticated, message is marked with DWS error and routed according to DWSLTT. {1:F01ABCDUS30AXXX1723226551}{2:O019 2206031027LWDXXXXX3XXX000012481603 10271706S}{4:{175:1421}{106:031023ABCDU S30AXXX1718294496}{108:FDC03102305723 00}{102:CSPBSGS0XXXX}{432:01}}{5:{CH K:960294864BC1}{SYS:1421031023ABCDUS 30AXXX1718294496}{TNG: }}

Abbreviation	Origin	Name	BMS Flows
			Note: Table continues from the previous page
APC-ISN	CBT	APC Input message (APDU 01 – A)	BMS gets GPA messages from data entry queue and routes them onto SWIFT GPA READY queue. DWSDGPA reads message from SWIFT GPA READY queue, adds necessary trailers (MAC, CHECK, etc), and ISN. If number of not-acknowledged messages is less than window size DWSDGPA sends message to SWIFT. Otherwise DWSDGPA waits for ACKs. If DWSDGPA does not receive an ACK for a message for 10 minutes, it aborts the GPA session. When a message is sent to SWIFT, it remains in SWIFT ready queue waiting for ACK. After session is restarted, If DWSDGPA finds a message in SWIFT READY queue which was already sent(it contains ISN and trailers), PDE trailer is attached. {1:A01ABCDUS30#XXX0000000000}{2:I043SWFTXXXXXXXX}{5:{CHK:4454D4405050}{TNG:}}
FIN-ISN	CBT	FIN input message (APDU 01 – F)	BMS gets messages from back-end applications or data entry and routes them on SWIFT READY queues (four queues for different priorities for each LT). DWSDGPA reads message from SWIFT READY queue (after processing each message it start reading from the queue with the highest priority), adds necessary trailers (MAC, PAC, CHECK, etc), and ISN. If number of not acknowledged messages is less than window size DWSDGPA sends message to SWIFT. Otherwise DWSDGPA waits for ACKs. If DWSDGPA does not receive an ACK for a message for 10 minutes, it aborts the FIN session. When a message is sent to SWIFT, it remains in SWIFT ready queue waiting for ACK.

Abbreviation	Origin	Name	BMS Flows
			After session is restarted, If DWSDGPA finds a message in SWIFT READY queue which was already sent(it contains ISN and trailers), PDE trailer is attached. {1:F01ABCDUS30#XXX0000000000}{2:I547U BSWGB20XEQUN}{3:{108:1032890051167}} {4: :16R:GENL :20C::SEME//1032890051167 :23G:NEWM :16R:LINK :20C::RELA//EFGH1S1YS1/001 :16S:LINK :16S:GENL :16R:TRADDET :98A::ESET//20031021 :98A::TRAD//20031016 :35B:ISIN US3825501014 GOODYEAR TIRE RUBBER COMPANY COM :70E::SPRO///DTCID/356981234 /ACOM/098 79100 :16S:TRADDET :16R:FIAC :36B::ESTT//UNIT/1800, :97A::SAFE//381234 :16S:FIAC :16R:SETDET :22F::SETR//TRAD :16R:SETPRTY :95Q::REAG//UBS SECURITIES LLC :97A::SAFE//00001234 :16S:SETPRTY :16R:SETPRTY :95Q::BUYR//UBS SECURITIES LLC :97A::SAFE//00641234 :16S:SETPRTY :16R:AMT

Abbreviation	Origin	Name	BMS Flows
			:19A::ESTT//USD12582, :16S:AMT :16S:SETDET - }{5:{MAC:801362D2}{CHK:1F0001B1C28A} {TNG:}}
FIN-ISN-ACK-NAK	FIN	FIN ISN acknowledgement message (APDU 21 – F)	BMS receives from SWIFT ACK on input FIN message which contains: 0. 177 date-time Date and time, local to the user, of the *Service Message 21 ACK/NAK.* 1. 451 accept-reject Accepted or rejected, where: • 0 = accepted • 1 = rejected 2. 405 rejection-reason Reason for rejection. BMS reconciles ACK with original message from send ready queue and route it according DWSLTT table {1:F21ABCDUS30AXXX1723307009}{4:{177:0310271804}{451:0}{108:CRS0310210000100}}
APC-ISN-ACK-NAK	FIN	APC ISN acknowledgement message (APDU 21 – A)	BMS receives from SWIFT ACK on input GPA message which contains: 1. 177 date-time Date and time, local to the user, of the *Service Message 21 ACK/NAK.* 2. 451 accept-reject Accepted or rejected, where: • 0 = accepted • 1 = rejected 3. 405 rejection-reason Reason for rejection. BMS reconciles ACK with original message from send ready queue and route it according DWSLTT table {1:A21ABCDUS30AXXX1631000001}{4:{177:0310271514}{451:0}}

Abbreviation	Origin	Name	BMS Flows
			Note: Table continues from the previous page
FIN-OSN-ACK-NAK	CBT	FIN OSN acknowledgement message (APDU 21 – F)	BMS sends to SWIFT ACK on output FIN message receiving from SWIFT. ACK contains: 1. 177 date-time Date and time, local to the user, of this message. 2. 451 accept-reject Accepted or rejected, where: • 0 = accepted • 1 = rejected 3. 405 rejection-reason Reason for rejection.
APC-OSN-ACK-NAK	CBT	APC OSN acknowledgement message (APDU 21 – A)	BMS sends to SWIFT ACK on output GPA message receiving from SWIFT. ACK contains: 1. 177 date-time Date and time, local to the user, of this message. 2. 451 accept-reject Accepted or rejected, where: • 0 = accepted • 1 = rejected 3. 405 rejection-reason Reason for rejection. {1:F21ABCDUS30AXXX1723226546}{4:{177:0310271626}{451:0}}
REMOVE-LT	FIN	emove LT message (APDU 14)	{1:A14ABCDUS30AXXX1623000003}{4:{443:002}}
REMOVE-AP	FIN	Remove APP message (APDU 12)	{1:A12ABCDUS30AXXX1624000004}{4:{443:006}}

Table 14 CBT – SWIFT FIN Interface

2.5 Message Control Point (MCP)

2.5.1 Hardware and Software Components

IBM mainframe hardware has excellent features that suit well for FI MCPs.

2.5.1.1 Features of the IBM z990 Processors

IBM's latest z990 "T-Rex" processors. The planned subsequent operating system upgrade to z/OS 1.4 will further allow to exploit the features of the new hardware.

The z990 has almost 3 times the capacity of the previous z900 series.

The maximum number of LPARs doubles from 15 to 30; eventually up to 60 will be supported.

2.5.1.2 Processor Basic Building Blocks

The z990 is based upon an architectural unit of a "book" which is a board containing processors, memory and connections to I/O cages. A z990 has from 1 to 4 books, with the designations of A, B, C and D for the different models.

Each book has 12 processors (PU's) not all of which may be activated.

- Up to 8 may be used as CPU's.
- 2 are spares for automatic fail over. In the event of a hardware failure on one of the CPU's a spare will automatically be activated to replace it.

- 2 are System Assistance Processors to handle I/O management offloading this task from the ones used for CPU processing.

At the top end of the line, a D32 can have up to 256 GB of memory. For example, the A08 model can have from 24 GB of memory up to 64 GB of memory. Memory is available in 8 GB increments.

2.5.1.3 I/O System

The I/O system can handle up to 96 GB/sec, which is 4 times the bandwidth of the previous z900. Individual channels can go up to 2 GB/sec.

The z990 provides much larger numbers of Fiber connections (FICON) to fully utilize the bandwidth and potential for dramatic improvements in I/O throughput.

2.5.1.4 Hardware Encryption

The z990 comes with new hardware encryption facilities, a "cryptographic coprocessor" PCIXCC.

Intelligent Resource Director (IRD) leverages the proven Workload Manager (WLM) to dynamically provide system resources to workloads that need them as per the Bank's defined priorities. It has three components :

- LPAR CPU management - distributes processor resources by dynamically adjusting LPAR weights as per workload requirements across the LPAR cluster.

- Dynamic Channel Path Management (DCM) - moves channel paths from one I/O control unit to another in response to workload.
- Channel Subsystem Priority Queuing - work that needs the I/O resources gets it as per priority, as opposed to default of FIFO.

The T-Rex processors can take more advantage of the IRD functions.

The Sysplex Distributor function of z/OS provides intelligent load balancing of TCP/IP traffic across a Parallel Sysplex cluster using Dynamic Virtual Internet Protocol Addressing (VIPA).

2.5.1.5 References

z990 home page
http://www-1.ibm.com/servers/eserver/zseries/990.html

Redbook - z990 Technical guide
http://www.redbooks.ibm.com/redbooks/pdfs/sg246947.pdf

Intelligent Resource Director (IRD)
http://www-1.ibm.com/servers/eserver/zseries/library/techbriefs/irdtechbrief.html

Redbook on IRD
http://www.redbooks.ibm.com/redbooks/pdfs/sg245952.pdf
Redbook - OS/390 Workload Manager Implementation and Exploitation

http://www.redbooks.ibm.com/redbooks/pdfs/sg245326.pdf
Software:
DB2 v7, CICS TS 2.2, Visual J++, Crystal Enterprise 9.0 (Windows)

2.5.2 Server Data Bases

The MCP contains multiple data bases that can be used in parallel. Each FI can have a unique data base. But the typical set of data bases is the following:

Customer Information Data Bases.

Bank Identification Codes.

Currency Codes.

Login and Authentication Keys.

User Profiles.

Message Queue Data Set.

Message Repository.

Message Format Standards.

Business Rules.

The Customer Information DB contains complete information about a customer, including:

- Full name and mailing address.
- Customer identification (for ex., 10-digit number) and name key (16 characters).
- SWIFT, FED, CHIPS, TLX addresses (if any).
- Fax and regular telephone numbers.
- Charging information (and 3rd party charging if any).
- Acceptance of select calls (messages), etc.

User profile contains access information for each individual user of BMS.

2.6 Operation, Administration and Maintenance

2.6.1 Operating System

Operating System depends on the hardware platform used. Example of hardware configuration is described below.

2.6.1.1 OS/390

- Processor: ESA/390, or compatible processor, that can run OS/390 Version 2.10, or higher, with enough processor storage

2.6.1.2 AIX

- Processor: 2-way 450MHz RS64 III
- RAM: 1 GB
- Disk space: 2 x 18 GB
- Display: Any graphical display that supports X-Windows

2.6.1.3 Windows 2000 or NT

- Processor: Intel Uniprocessor, RAM: 512 MB

2.6.1.4 Example of the corresponding software:

- OS/390 2.10 (5647-A01) or z/OS 1.1 or higher
- AIX 4.3.3
- Windows 2000 SR 1 or Windows NT 4.0 SR 6

The following indicates the possible software of the middleware products:

- z/OS
- IBM SMP/E for z/OS and OS/390 3.1 (5655-G44)
- DB2 XML Extender V7
- REXX (part of OS/390 2.10)

- XML Toolkit for OS/390 1.3 (5655-D44)
- IBM WebSphere MQ Integrator for z/OS 2.1 (5655-G97)
- IBM WebSphere MQ for z/OS 5.2 (5655-F10)
- DB2 Universal Database for OS/390 7.1 (5675-DB2)
- IBM C/C++ Compiler 2.10 (5647-A01)
- AIX
- DB2 XML Extender V7
- Object REXX for AIX 1.1.2.0
- Visual Age for C++ for AIX 5.0.2
- IBM WebSphere MQ Integrator for AIX 2.1
- IBM MQ Series for AIX 5.2
- DB2 Universal Database for AIX 7.1
- Windows
- WMQI Version 2.1 for Windows
- Adobe Acrobat Reader (for documentation purposes).

2.6.2 Service Management System

SMS is the support facility to enter and maintain customer data in MCP data basses. Maintenance includes:

- Updating the data.

- Off-loading (daily) the archival messages on the long-term storage.
- Contingency.
- Upgrading DBMS.

Example of SMS function selection screen is shown on Figure 62.

```
                Function Selection                    Page 1

To select a function, move the cursor to ">" and press ENTER

     > CMD      Operator Command Processing
     > USR      User File Maintenance
     > AUT0     Authenticator Key File Maintenance / Display only
     > FLM      General File Maintenance
     > MSC      BMS System Control
```

Figure 62 Example of SMS Function Selection Screen.

2.6.3 Data Base Administration

DBA selects appropriate data bases, defines the structure, supports the initial load, maintains DB, image copies, disaster/recovery. In SQL-type DB (DB2, WEB Logic) DBA defines table spaces and their relations, partitions (4-64Gb), scalability, optimization for better performance.

2.6.4 Service Creation Environment

New SCE language for FIIN facilitates location-application combinations as set of macros and standard naming convention for message hops (queues).

SCE coding example of application entry:

```
MRVBRAFT APL=GPS,ITF=MQA,C1=Y,
PO=(D,D,D,D,D,D,D,D,D,D,Y,Y),
PI=(D,Y,D,D,D,D,D,D,D,D,D,Y),
BO=(Y,Y,Y,Y,D,Y,Y,Y,Y,Y,Y,Y),
QO=(Y,Y,Y,Y,D,Y,Y,Y,Y,Y,Y,Y),
DE=(Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,Y),
BRC=(FRP,GB2,BEB,DEF,JPJ,LUL)

MRVBRAFT APL=BNE,ITF=MQN,C1=D,
IO=(N,N,N,N),
BRC=(GBE,JEE,LUE,EOM)
MRVBRAFT APL=ATL,ITF=NDM,
PO=(Y,D,D,D,D,D,D,D,D,Y),
PI=(Y,D,D,D,D,D,D,D,D,Y),
BO=(Y,D,D,D,D,D,D,D,D,Y),
DO=(Y,D,D,D,D,D,D,D,D,Y),
DE=(Y,Y,Y,Y,Y,Y,Y,Y,Y,Y),
BRC=(CNS,DEF,SGS,HKH,JPJ,KRS)
```

2.6.5 Service Administration

Service Administration controls Message Flow and access to the services provided by FIIN. It controls the following:

- Prioritization mechanism, troubleshooting, error correction and distribution of message flow.
- Monitoring procedures for standardized message queues.

- Scalable solution for increasing traffic and throughput.

Example of Operator Command Processing:

DQ FRPGPS

DSL143I Display Queues

Function	K	USR	WAIT	THRSH	Function	K	USR	WAIT	THRSH
FRPGPSIH	N	3	00 000000	00100	FRPGPSDO		0	00 000000	00000
FRPGPSQO	N	0	00 000000	00020	FRPGPSKO		3	00 010044	00000
FRPGPSJO		3	00 000000	00000	FRPGPSCI	N	0	00 000000	00020
FRPGPSKI		3	00 002214	00000	FRPGPSJI		3	01 000001	00000
FRPGPSAI		3	00 000005	00000	FRPGPSEI		3	03 000002	00000
FRPGPSTI		3	00 000019	00000	FRPGPSVI		3	06 000000	00000
FRPGPSBI		3	00 000094	00000	FRPGPSLI		3	00 000000	00000
FRPGPSPI	A	0	00 000000	00000	FRPGPSBO		3	00 000179	00000
FRPGPSPO	A	0	00 000000	00000					

143952 is the time of this display
BMP1 Command ======>
PF 1=Help 2=Repeat 3=Return 4=DF 5=DU 6=DM Last
PF 7=Page -1 8=Page +1 9=Retrieve 10=DP 11=DQ filled 12=DL

2.6.6 User Access Administration

Access administration main concerns are confidentiality and accountability. Information is disclosed to only authorized personal/applications at the authorized locations. Every individual/application trying to gain or gaining the access to information is accountable.

User profile defines each individual user access to BMS. Privileges can include the following:

Network selection access (ability to LOGIN, etc.).

Origin ID (for ex., SWIFT address).

Function selection.

Security administration functions (if any).

List of authorized message types.

List of restricted commands.

2.6.7 System Access Administration

System access is granted to privileged users only. If system (control) information is separated from the data (messages), then there is no need for business users to interfere with the system functions. FIIN have a built-in logical separation of data messages from system information (including different acknowledgements, confirmations, etc.).

Software system administration is similar to SS7.

2.7 Data Security and Integrity

The current discussions are revolving around using Public Key Infrastructure (PKI) with phased migration from USE - User Security Enhancements consisting of Secured Login and Select Keys (SLS) and Bilateral Key Exchanges (BKE).

BKE prevents exchanging the correspondence between parties who did not yet set up the mutually-agreed relationship by exchanging the keys first. In PKI environment a sender can generate the traffic unwanted by the recipient. This traffic has to be stopped either centrally by a network or locally by a recipient. The solution is expected by mid-2004.

Overall concept of secured access to financial network is shown on Figure 63.

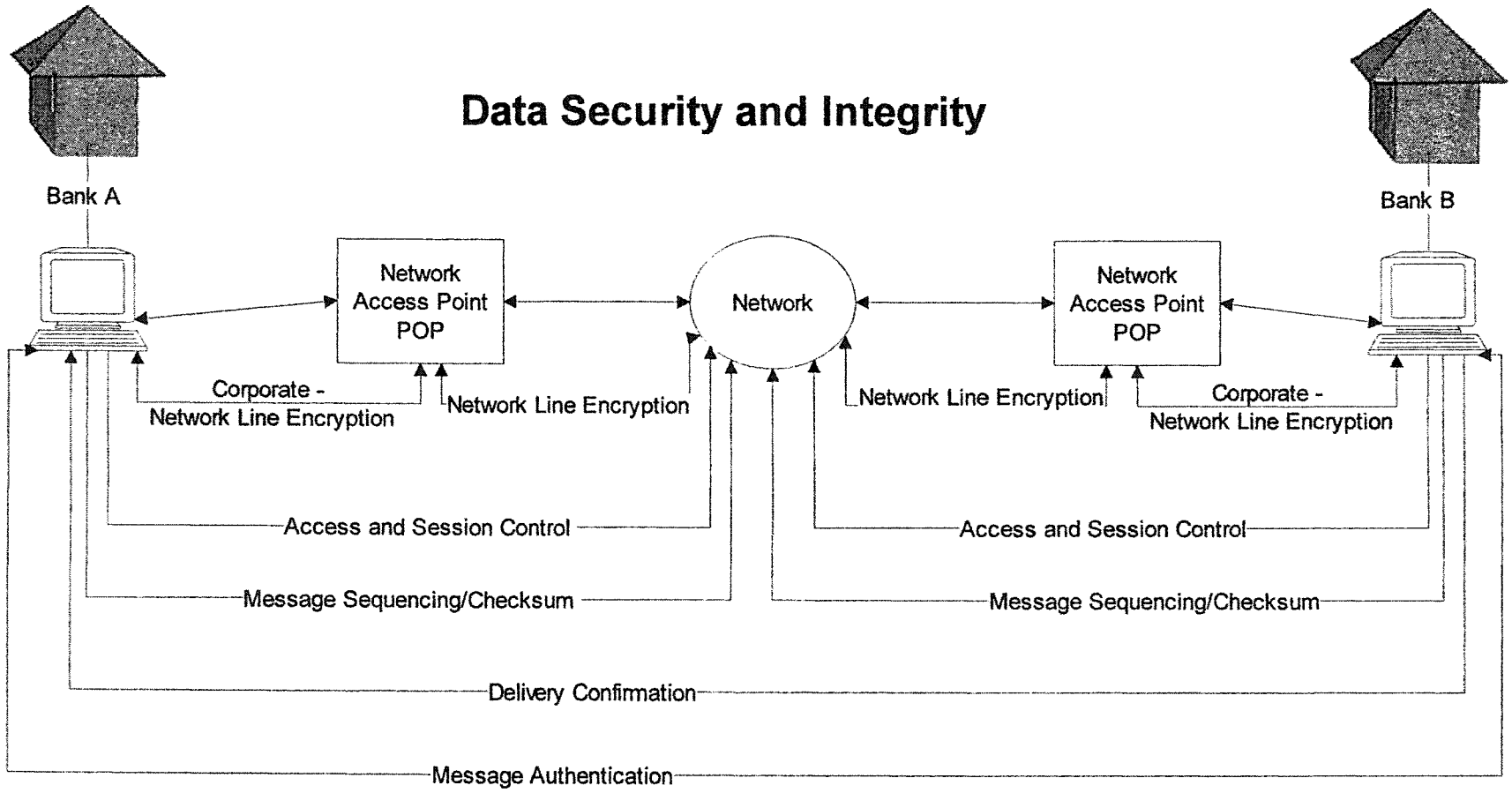


Figure 63 Secured Network Access.

2.7.1 Line and Data Encryption

Financial networks use line and data encryptors such that information never travels through the network in clear. For ex., SecureX25H (high speed) encryptor.

2.7.2 Network Login Key Infrastructure

LOGIN and SELECT keys can be generated by 'smart' Integrated Circuit Cards (ICC) in tamper-resistant Secure Card Reader (SCR). SWIFT uses proprietary USE system. Example of the keys see in Table 14.

2.7.3 Message Authentication

The authentication allows the receiver to check the integrity of a message and verify the identity of the sender independently of a network. CBT of a sender calculates a Message Authentication Code (MAC) trailer of the message using the appropriate key and network-provided algorithm (for ex., SA-2). The receiver's CBT automatically verifies the MAC trailer by recalculating it using the appropriate key and the same algorithm. The same MAC trailer proves the message origin and integrity.

Checksum trailer strengthens the secure delivery of a message with MAC trailer. Checksum is the only trailer for informational (non-transactional) messages.

2.8 Message Control System (MCS)

Security is a central and essential aspect of FIN. It protects from malicious or fraudulent use of the network, minimizes the vulnerability of physical resources to the consequences of unauthorized access or environmental disaster. MCS is responsible for the security and reliability of all services including non-financial internal and transmission and delivery of financial messages.

To establish a session MSP accepts LOGIN command from authorized operator and then sends the request for LOGIN key via MTP to MCP. MCP retrieves the next available key from Secure LOGIN Keys data base and forwards it to MSP to complete the LOGIN. All keys are encrypted and cannot be displayed. After session has been established MCS controls all incoming and outgoing traffic.

For each outgoing message MSP sends a query via MTP to MCP for receiver's address translation on base of its 10-digit identification (CID) or 16-character namekey. MCP looks up the CIF and BIC data bases and returns all available addresses for that subscriber (Telex, SWIFT, cable, Fax, mailing address). MSP then makes the routing decision and sends the message. MCS is looking for acknowledgement, positive or negative, for every outgoing message and acts accordingly.

For each incoming message MCS validates the sender and tries to authenticate the message. Again MSP requests via MTP from MCP the authenticator key for the sender of the message.

MCP is the end point in MCS to provide a quick response to queries from various MSPs. This information is crucial to complete the service requested.

SMS serves MCP via the localized link.

Message validation ensures compliance with syntactical and conditional rules of message formats including the presence of mandatory fields, proper tag field sequence order, the appropriate use of the message header and trailer blocks. It allows message recipient the straight through processing without manual intervention.

All messages are assigned unique input and output sequence numbers upon entering and exiting MCS. If number received is not the one that expected, then there two cases: gap or duplicate. MCS reacts depending on a protocol with a subscriber: warning or abort the session.

2.8.1 Message Control Link

One of the problem of any FIN is the secured access to a carrier. SWIFT, for example, as more advanced PVN, offers two systems as a base for

customization by FIN: User Security Enhancement (USE) and Bilateral Key Exchange (BKE) [21].

The random and response key tables (for LOGIN and SELECT) are not transmitted through the network. Instead, they are used by FIN and SWIFT in every access procedure, to generate unique security codes which are exchanged and verified by both parties. USE subsystem provides smart card reader technology at all users' sites for system selected LOGIN keys each time a user logs into the system: Secured LOGIN Select (SLS).

FNs exchange authenticator keys by using BKE subsystem.

Message Authentication Code (MAC) is created at input time based on the message contents.

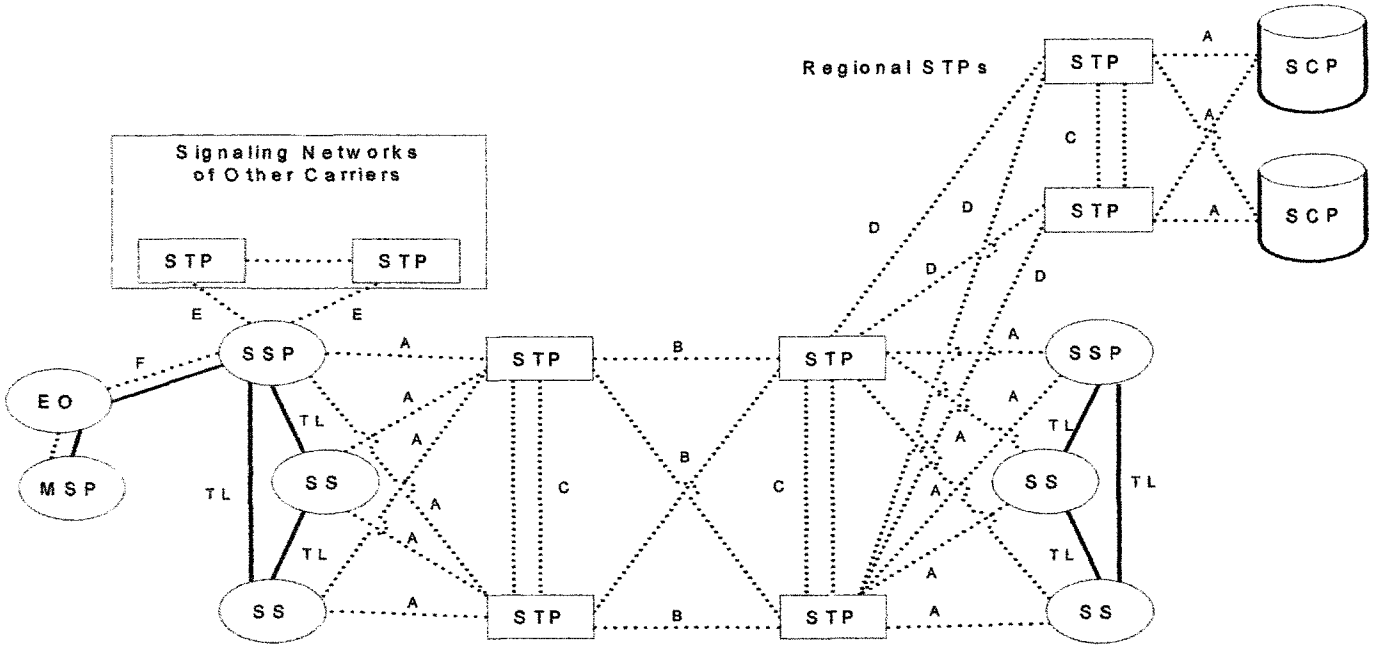
All together MAC, SLS, BKE ensure that messages are secured from alterations (deliberate or accidental) during the transmission process.

For confidentiality of the message contents SWIFT provides the full data encryption. Since messages are stored in encrypted form, even SWIFT personal cannot read them unless specific authorization for decryption has been obtained from the sending institution.

SWIFT provides financial guarantee for delivery of all messages that are positively acknowledged and accepted by SWIFT. It is a unique feature among global network services providers. SWIFT assumes financial liability for direct and interest losses of authenticated payments or transfers within SWIFT's area of control.

The SS7 network exchanges signaling information to control and monitor INs. The transition from in-band, multi-tone, or trunk signaling in POTS to Common Channel Signaling System 7 (CCS7) in INs was gradual. The CCS7 used out-of-band information to control and signal the various switches to complete and monitor the information bearing channels [**2**, **3**, **4**].

The interconnection of STPs in the CCS7 Signaling Network and MSP integration into it is depicted on Figure 64. SSP is connected to STPs of other carriers through the Extended links, and connection to SSP is done through Fully-Associated links [**1**, **2**].



SCP = ServiceControl Point; STP = Service Transfer Point; SSP = Service Switching Point; SS = Switching System; EO = End Office; TL = Trunk Lines; A = Access Link; B = Bridge Link; C = Cross Link; D = Diagonal Link; E = Extended Link; F = Fully-Associated Link, MSP = Message Swithcing Point.

Figure 64 MSP Integration into CCS7 Signaling Network.

Control information monitors a message flow through FIN similar to monitoring voice/data transfer in IN [2]. Circuit, packet, or channel switched networks can be used for transport of messages [3, 4].

System and administrative messages control and monitor the flow of financial messages. Control information can be logically separated and monitor the financial message flow over proposed Message Control System (MCS).

All messages are assigned input and output sequence numbers upon entering and exiting SWIFT network. These numbers are verified during sending and reception for each and every message. If they do not follow the expected sequence, the message is not only rejected, but the terminal session is also aborted. Renewed LOGIN by the user is then required thereby enforcing strict accountability control.

User usually requests the "best method" for delivery of a financial message. And it is for the routing intelligence modules of FIN to decide what is the "best" in transferring of each particular message at any given time. The possible routing criteria are the following:

- type of service subscribed by sender and recipient (SWIFT, Telex, FAX,
- cable, regular mail);
- availability of a carrier;
- source and destination of information;
- minimal cost;

- minimal delay;
- reliability.

There are 9 different categories of financial messages described in SWIFT/ISO 7775 message standards and adapted by ISITC (Industry Standardization for Institutional Trade Communications) [**21**]:

1. Customer Payments and Checks.
2. Payments, Cash Management and Customer Status.
3. Financial Trading.
4. Collections and Cash Letters.
5. Securities.
6. Precious Metals and Syndications.
7. Documentary Credits and Guarantees.
8. Travelers Checks.
9. Cash Management and Customer Status.

The rigid message format with standard tags and codes instead of free format text allows automatic message processing by applications. Straight through processing eliminates costly and error-prone manual processing.

For message transmission error control a mandatory checksum is added to all financial messages. It enables a receiver of a message to verify that a message text has not been corrupted due to system malfunction or undetected transmission error.

Message and packet switching techniques are used to process large volumes of data over the networks [9, 10, 11]. With circuit switching, the physical circuits (or transmission paths) are switched. With message and packet switching, circuits remain permanently connected and messages themselves are switched. Message switching can operate in non-real time communications in store-and-forward mode when one party is not connected. Messages wait delivery until recipient accesses the network. Messages are also kept on history queues for possible later retrieval. By contrast, networks normally do not retain packets once they have been delivered correctly.

There two types of packet switching:

- multipacket messages. Requires packet assembly-disassembly function (PAD) that increases overhead;
- single-packet messages (sometimes called datagrams). Does not require PAD function. Single-packet messages do not exceed the maximum capacity of one packet.

Comparison of typical voice call, ISDN and FIN connections are shown in the Table 15.

Number	Typical Voice Call	ISDN Data Call	Network Connection
1.	Accept the complete sequence of digits (called party telephone number).	Accept the recipient number.	Accept LOGIN and authenticate the subscriber by matching the selected secured LOGIN key.
2.	Complete ringing connection.	Identify an idle B channel over the D channel.	Dedicate the line for the session.
3.	Supply ringing signals by alerting called and calling parties.	Set up the connection on the B channel.	Set up the connection over the line.
4.	Direct response on the channel.	Direct response on the D channel.	Detect the LOGIN acknowledgment.
5.	Disconnect the ringing path.	Receiver ready.	Receiver (line) ready.
6.	Establish voice path.	Data transfer over the B channel.	Messages transfer.
7.	Await hang up by either party.	Await release-suspend by either party.	Await LOGOUT or QUIT command.
8.	Disconnect the voice path.	Disconnect the B channel.	Detect LOGOFF or QUIT acknowledgment.
9.	Release the connection.	Release the channel to become idle.	Release the line.

Table 15 Comparison of Typical Voice Call, ISDN Data Call, and FIN Connection.

2.8.2 Message Link

After control link is established, i.e. CBT for particular Logical Terminal is connected to a Network, Data link can start sending/receiving financial Data messages with the connected Network.

2.8.3 Administrative Data Link

System maintenance takes place at administrative data link. For example, establish a relat6ionship with a new correspondent, replace the expired keys with the new ones, etc. This is similar to CCSS7.

2.8.4 Separation of Control and Data Messages

As shown in message flowcharts the input to network and output from network financial messages are separated from control messages. Acknowledgements coming from network on I-type messages are fully reconciled after being received by back-end application originated the I-type message. Acknowledgements on O-type message are sent to network and also reconciled after receiving a confirmation from back-end application.

2.8.5 Control Channel

MSP communicates with MCP data bases via MTP logical control channels to receive the information necessary to complete the routing of a message.

In the environment with multiple BMS instances (large FIs) the communications between multiple MSPs takes place via control channel which on mainframe is usually LU 6.2 connection between CICS regions. The similarity with CCISS7 can be observed.

2.8.6 Line Utilization

Multiple lease lines can connect each BMS instance to a network. The main concerns are the following:

- Flow Control and Bottlenecks.
- Line Allocation and Traffic Distribution.
- Contingency Handling (Switch to Backup Line, etc).

Research was done on rate of acknowledgments received from SWIFT using X.25 protocol over 56 kbps data send-only line with session layer window 30. FIIN sends 30 financial messages regardless of their size (up to 10,000 bytes) before it stops awaiting acknowledgement from SWIFT. SWIFT acknowledges each message separately on the same line. No data messages are

received over this line. FIIN later reconciles each acknowledgment with its original message.

Acknowledgement rate is shown on Figure 65.

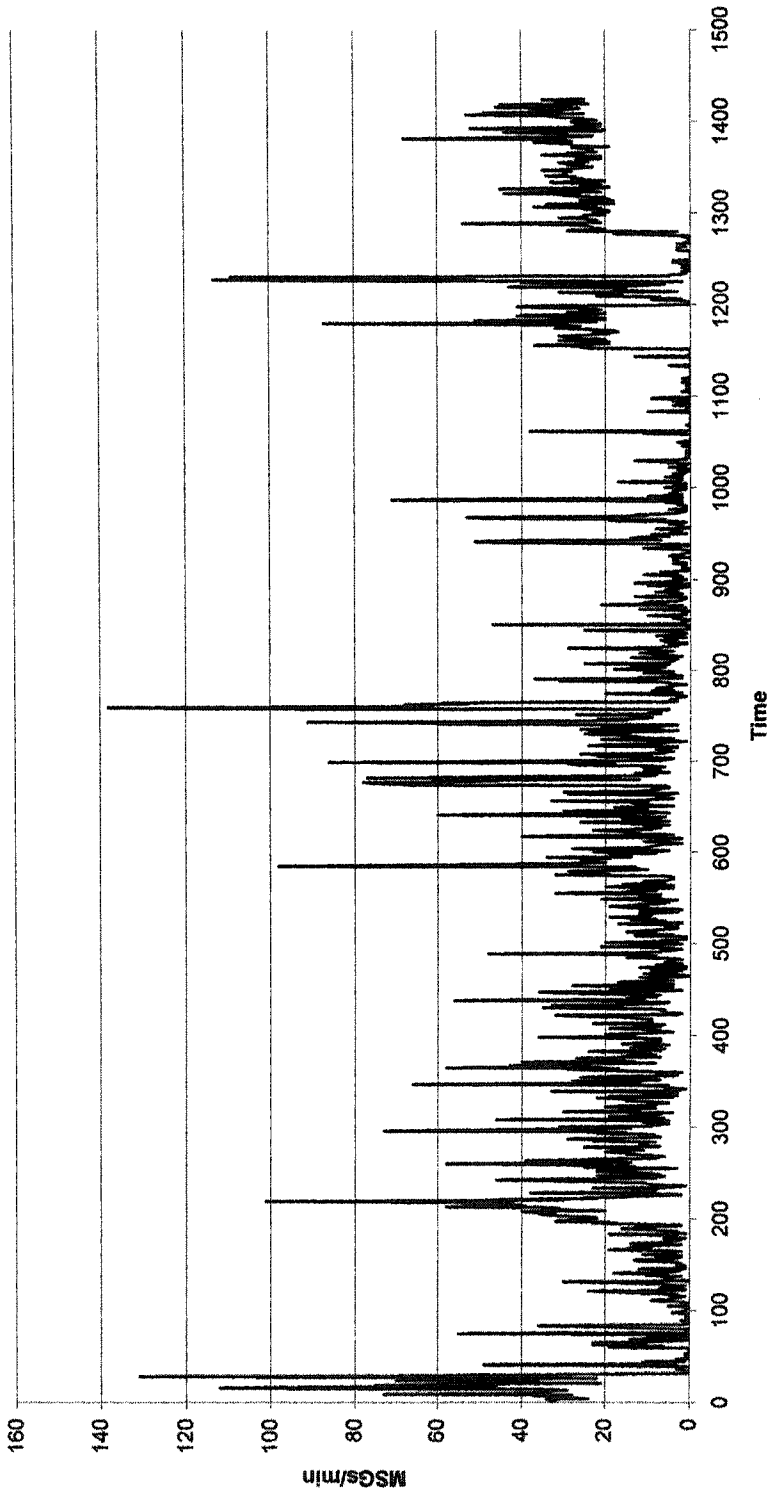


Figure 65 Acknowledgment Rate.

3 Interface Between Financial Institution Intelligent Network and Carriers

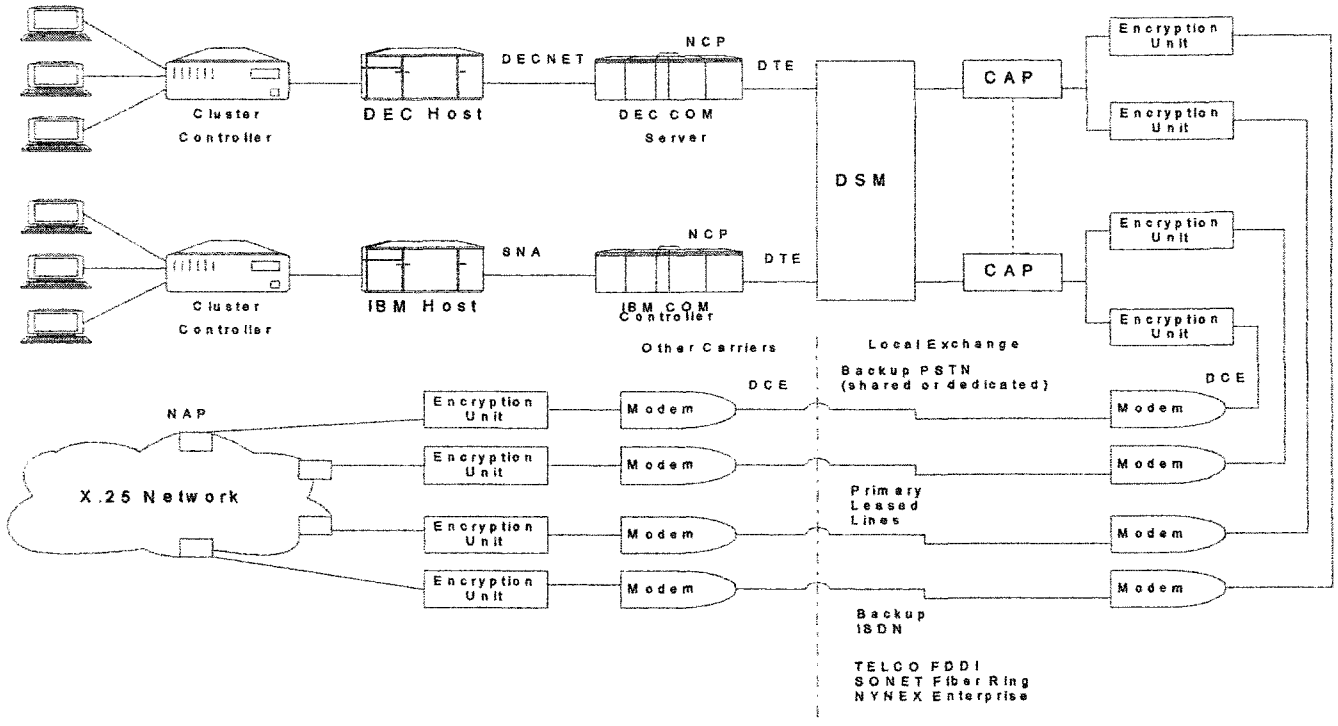
3.1 Private and Public Carriers

Connection to the carrier from multiplatform environment is more complicated. Simplified version of dual line FIN connection to NAP is shown on Figure 66.

In connection with public carrier two permanent virtual circuits are usually established: for outgoing and incoming messages (see Figure 67). Acknowledgement from the carrier or the recipient can be requested. X.25 full duplex protocol is used.

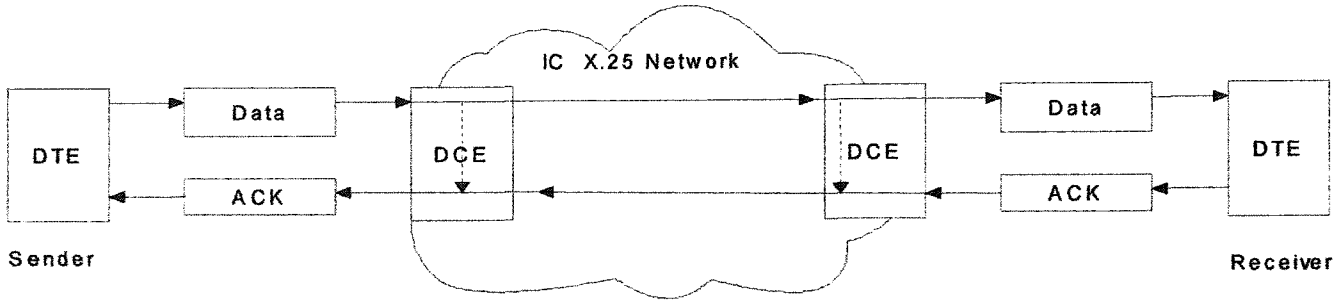
Detail of Single Window Concept connecting FIIN to financial networks using different protocols is shown on Figure 68.

Normal as well as variations of failures and recovery scenarios are shown on Figure 69, Figure 70, Figure 71.

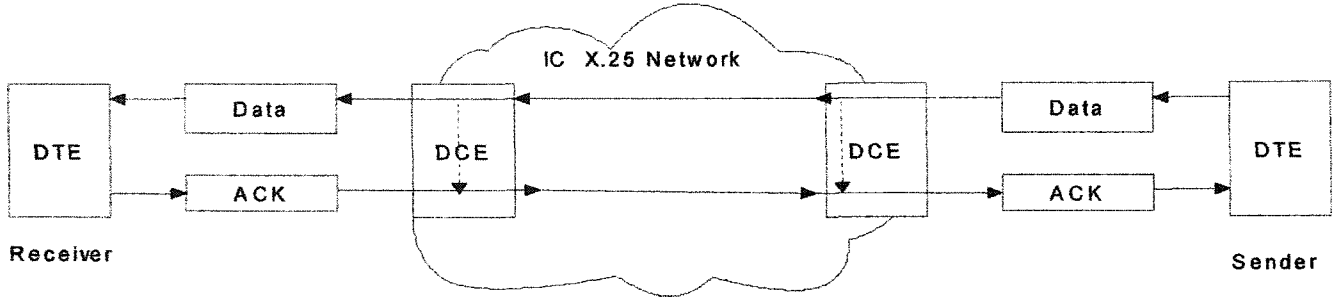


DECNET = Digital Equipment Corporation Network; SNA = System Network Architecture;
DTE = Data Terminal Equipment; DCE = Data Circuit-Terminating Equipment;
DSM = Digital Switching Matrix; CAP = Customer Access Point; NAP = Network Access Point;
PSTN = Public Switched Telephone Network; NCP = Network Control Program;
ISDN = Integrated Services Digital Network; FDDI = Fiber Distributed Data Interface;
SONET = Synchronous Optical Network.

Figure 66 Dual Line Connection to NAPs.



(a) PVC with LCN = 1 for Outgoing Messages.



(b) PVC with LCN = 2 for Incoming Messages.

IC = Interexchange Carrier; DCE = Data Circuit-Terminating Equipment;
DTE = Data Terminal Equipment; ACK = Acknowledgement;
PVC = Permanent Virtual Call; LNC = Logical Circuit Number.

Figure 67 PVC Data Flow.

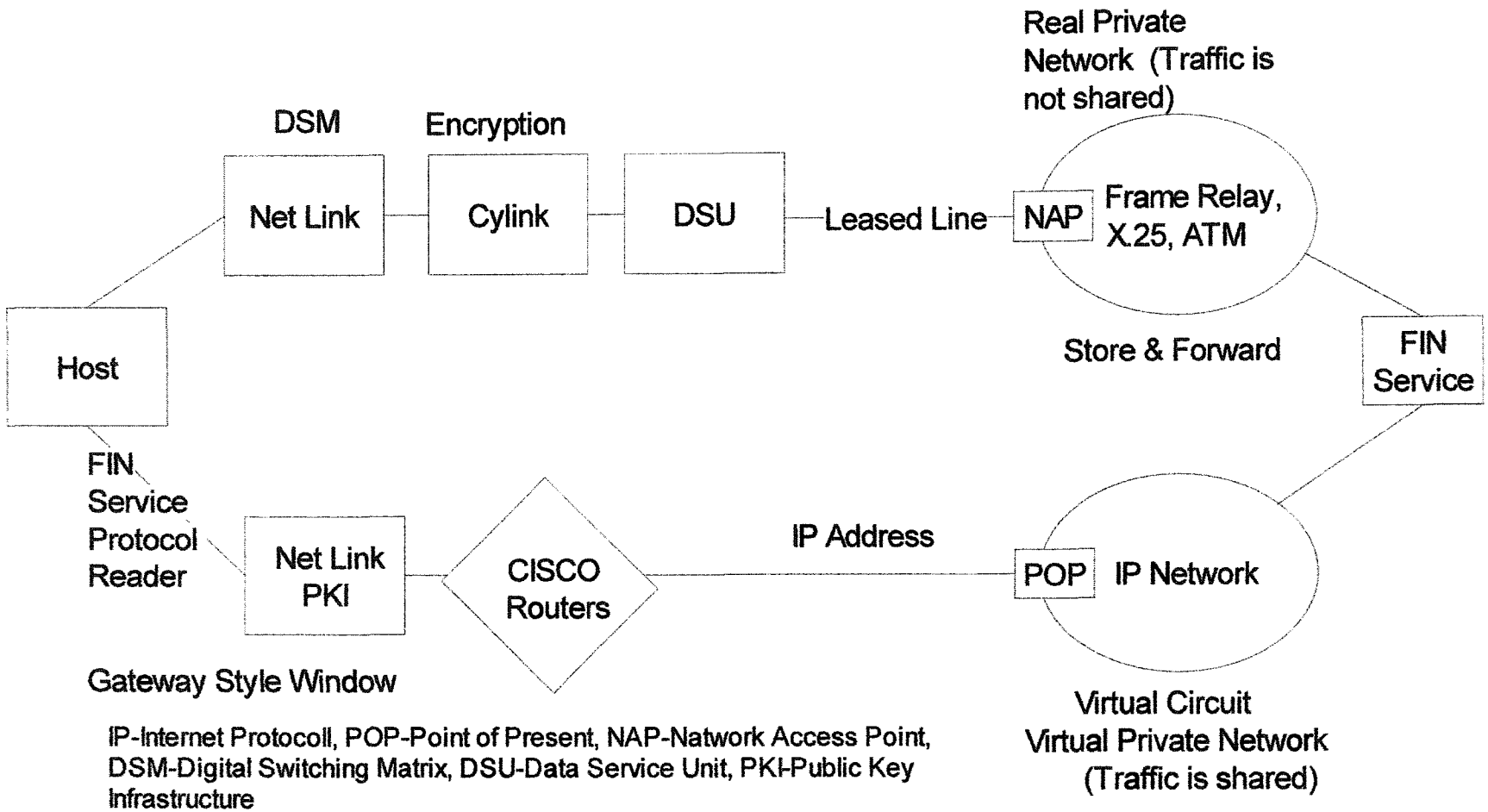


Figure 68 Single Window Detail.

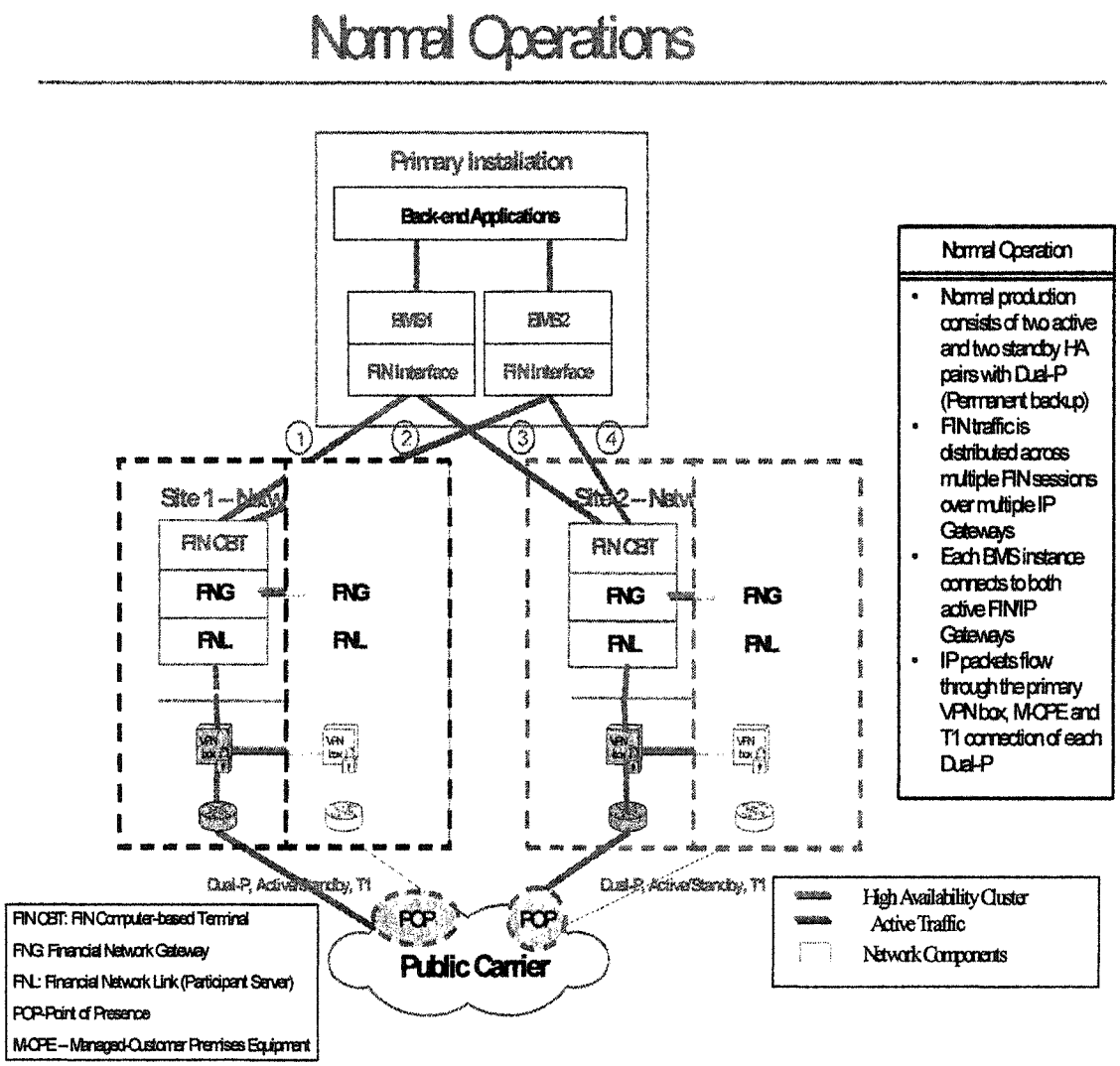


Figure 69 MSP Normal Operations.

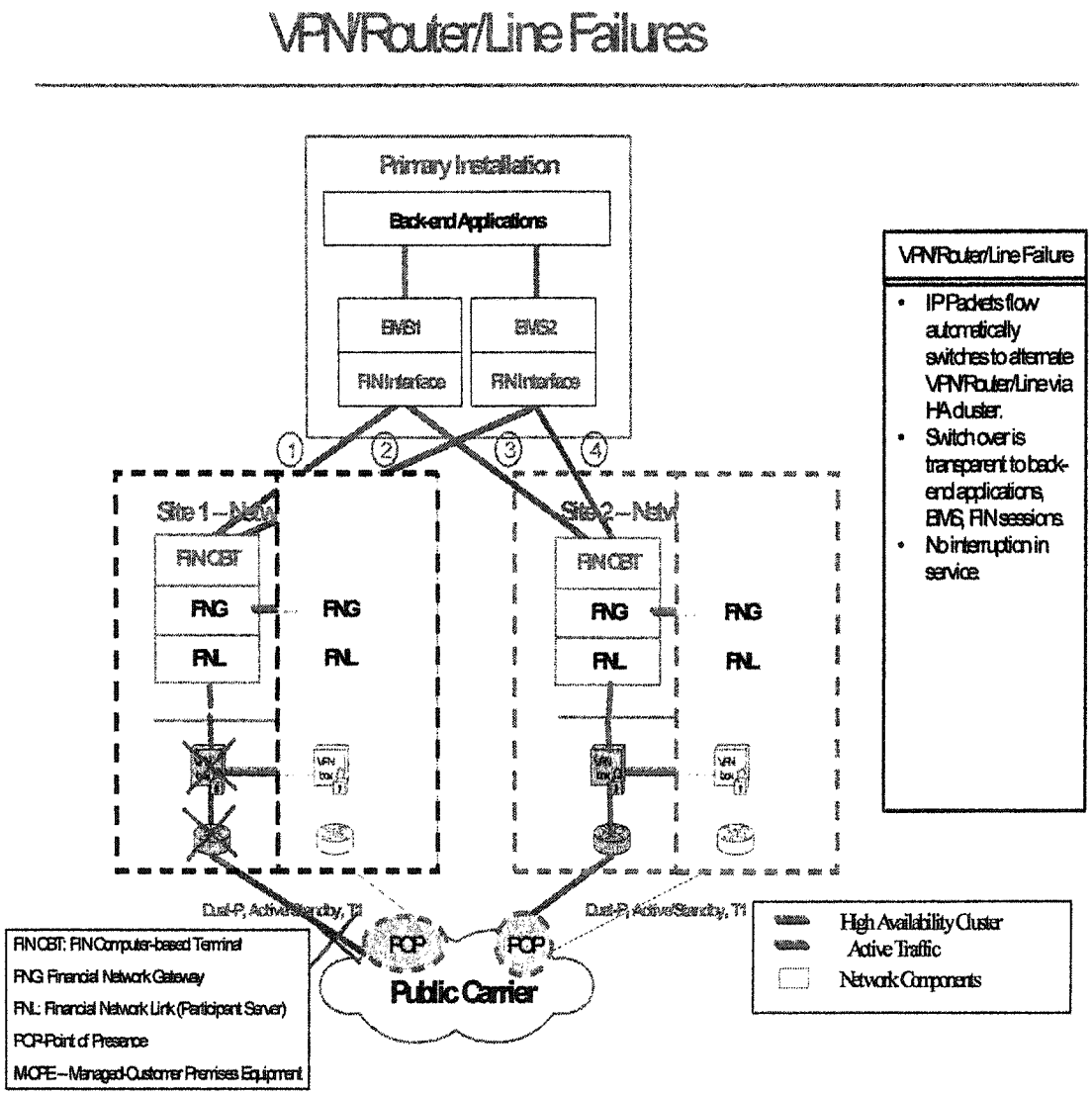


Figure 70 MSP Switch in Case of VPN/Router/Line Failures

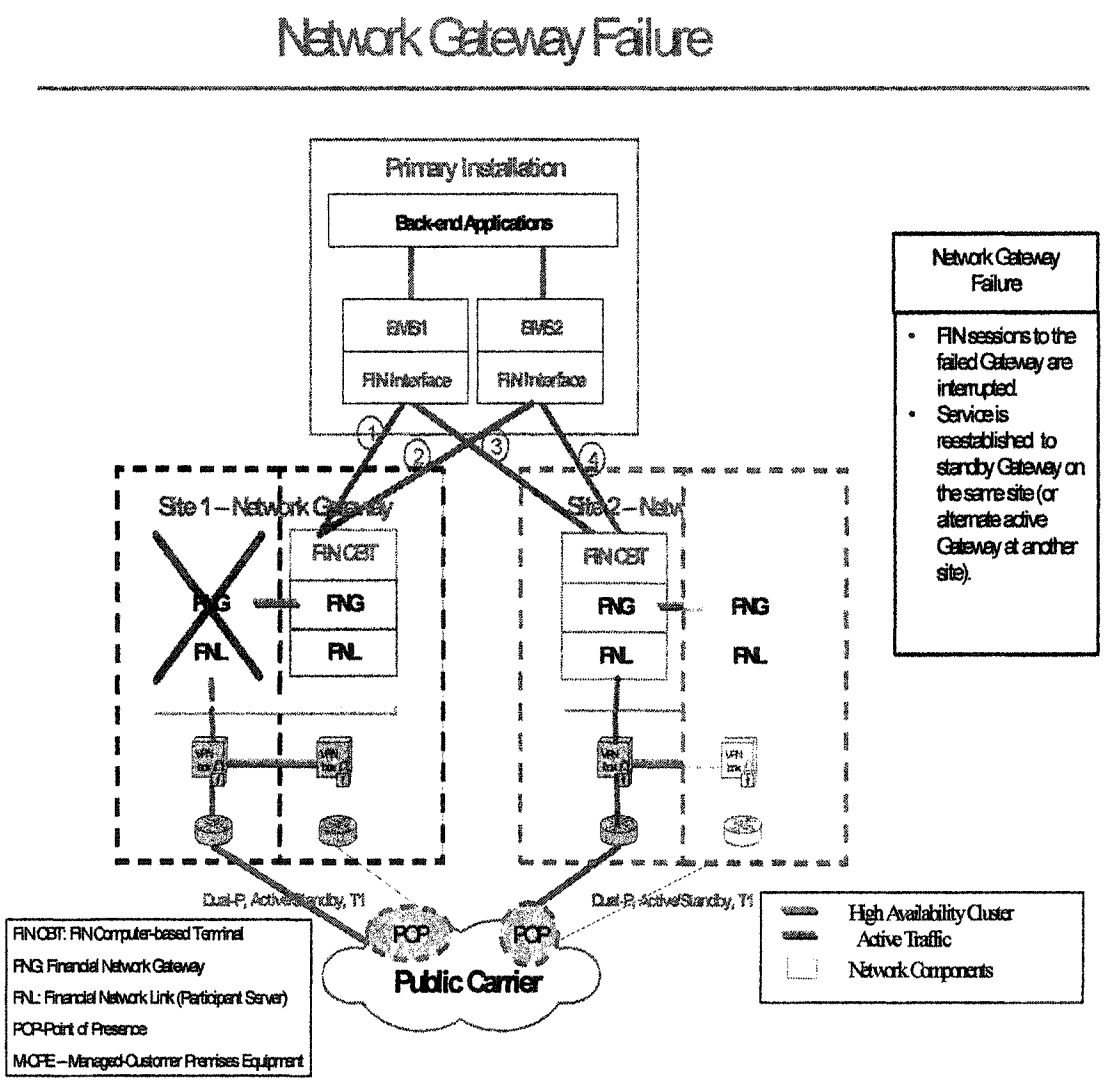


Figure 71 MSP Switch in Case of Network Gateway Failue

Direction/ Protocol	Command	BMS Mandatory	CBT Optional
I, LTC	LOGIN APDU 02	Lterm Login session number Login session key APC window size (=1)	MAC trailer (calculated on the base of login session key)
	LOGOUT APDU 06	Lterm	
	ABORTLT APDU 35	Lterm	
I, APC	SELECT APDU 03	Lterm Select session key FIN window size Select mode I, O, I/O Select subsets	MAC trailer (calculated on the base of login session key)
	QUIT APDU 05	lterm lterm	
	ABORTAP APDU 33		
I-ACK, LTC	LOGIN-ACK APDU 22	Status of Login LSN	MAC Authentication (based on LSN) Receive date/time LSN open/closed Last ISN received/ OSN sent
	LOGIN-NAK APDU 42	Set status to LOGOUT LSN	MAC Authentication
	ABORT-ACK		
	ABORT-NAK		
I-ACK, APC	SELECT-ACK APDU 23	Set status to OPEN	MAC Authentication

Direction/ Protocol	Command	BMS Mandatory	CBT Optional
Note: Table continues from the previous page			
	SELECT-NAK APDU 43	Set status to CLOSE	MAC Authentication
	QUIT-ACK APDU 25	Set status to CLOSE	MAC Authentication
I-ISN, APC	GPA APDU 01-A	MAC Trailer CHECK Check window size (abort after 10 min) PDE (msg is BUSY until ACKed)	ISN
I-ISN, FIN	FIN APDU 01-F	MAC CHECK Check window size (abort after 10 min)	ISN
I-ISN-ACK/NAK, APC	GPA 21-A	Reconcile with Original	Received date/time
I-ISN-ACK/NAK, FIN	FIN APDU 21-F	Reconcile with Original	Received date/time
O-OSN, APC	GPA APDU 01-A	MAC authentication - error if not authenticated	Check OSN abort if out of order
O-OSN, FIN	FIN APDU 01-F	MAC authentication - error if not authenticated	Check OSN abort if out of order
O-OSN-ACK/NAK, APC	GPA APDU 21-A	Local date/time NAK reason	
O-OSN-ACK/NAK, FIN	FIN APDU 21-F	Local date/time NAK reason	

BMS Bank Messaging System
LTC Logical Terminal Control
APC Application Control

APDU	Application Protocol Data Unit
LSN	Logical Session Number
SSN	Select Session Number
Window Size	Maximum number of messages sent or received without waiting for or sending acknowledgement
Lterm	Logical Terminal (for ex. BANKUS30A)
FIN	Network Financial Application
ISN	Input Sequence Number
OSN	Output Sequence Number
ISW	Input Sequence Window (not AVKed messages by Network)
OSW	Output Sequence Window (not AVKed messages by BMS)

Table 16 BMS Communications Primitives.

3.2 Transmission Lines

There are shared, dedicated, and leased lines.

Connection to POP is discussed in [23] and shown on Figure 72. Charges related to connection to POP are shown in Table 17.

Customer Access Point (CAP) switches to back up line if leased line fails, and switches back if leased line becomes available. Intelligent CAP can automatically switch to another CAP in its domain.

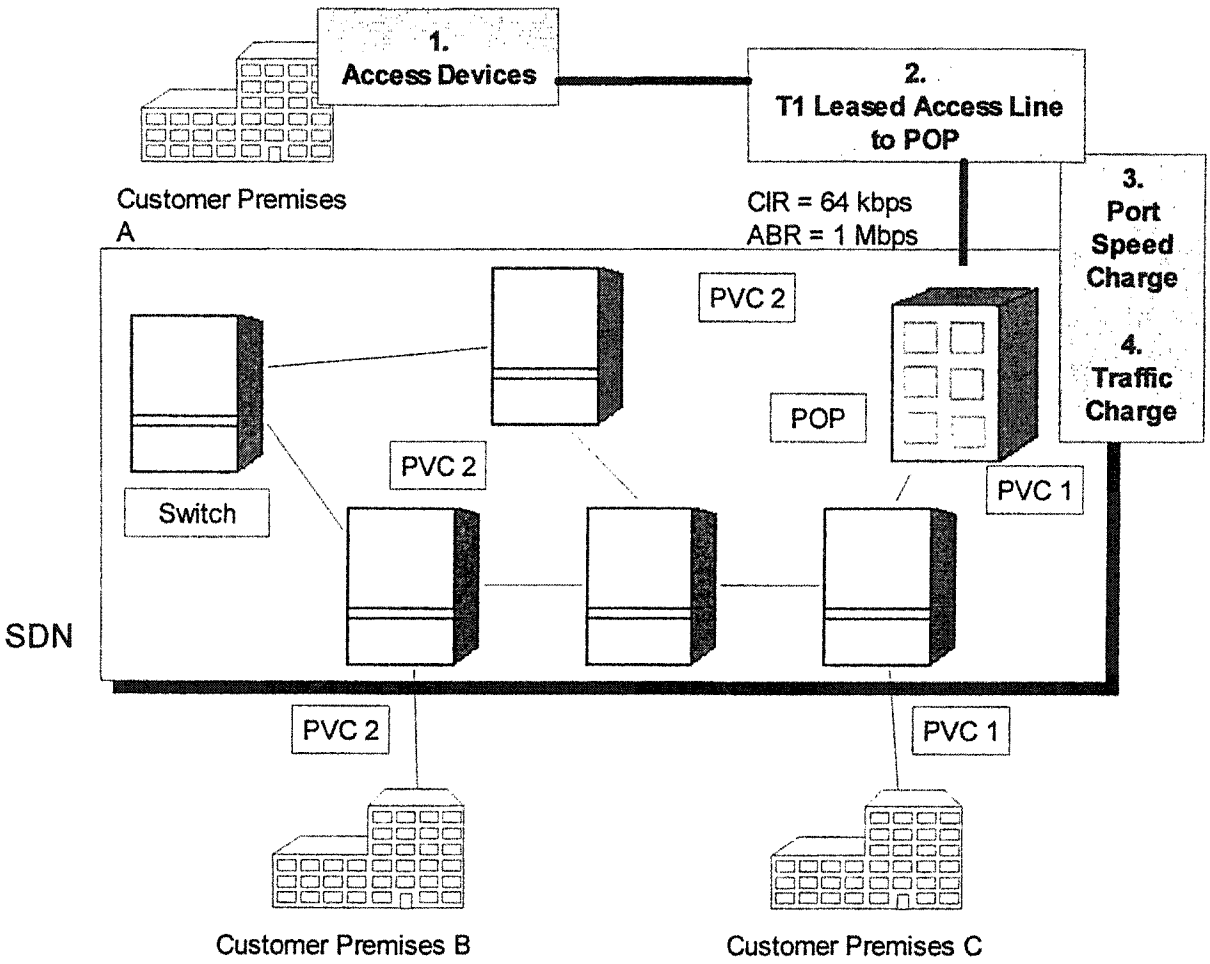


Figure 72 Connection to POP.

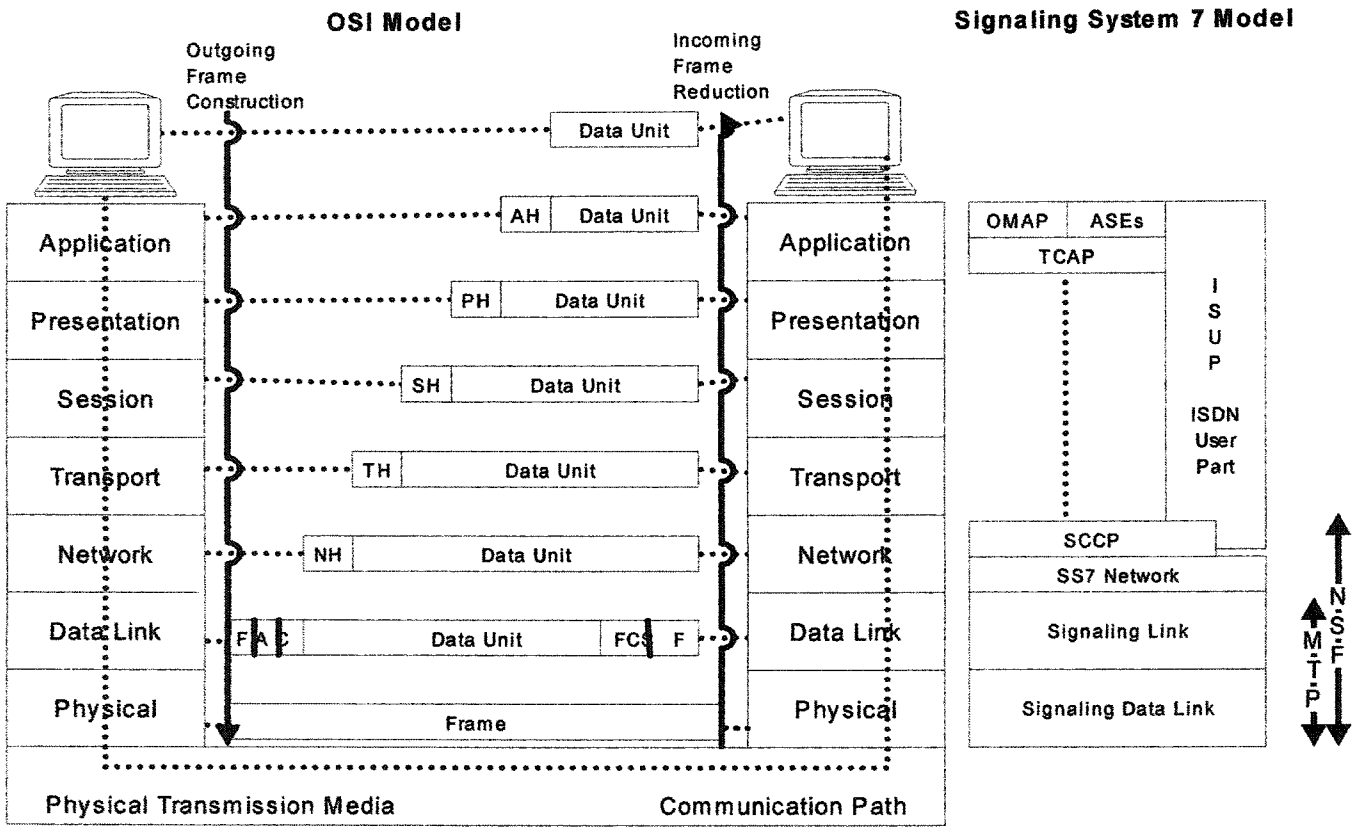
Elements	Vendor	Detail
Access Device	Hardware Vendor	Hardware to connect a corporate site network to the first switch through the leased access line
Leased Access Line	Telephone Company	To connect a corporate to the switched data network
		Setup charge and monthly charge; sometimes per-megabyte charge
Port Charge	PSDN Carrier	Single port speed charge
		Based on both committed information rate and available bit rate
Traffic Charge	PSDN Carrier	Charge per megabyte transmitted
PVC Charge	PSDN Carrier	Charge per permanent virtual circuit; may include speed and traffic charges
SVC Charge	PSDN Carrier	Charge per SVC is systems that offer switched virtual circuits

Table 17 Charges Related to Connection to POP.

4 Protocols in Financial Institution Intelligent Networks

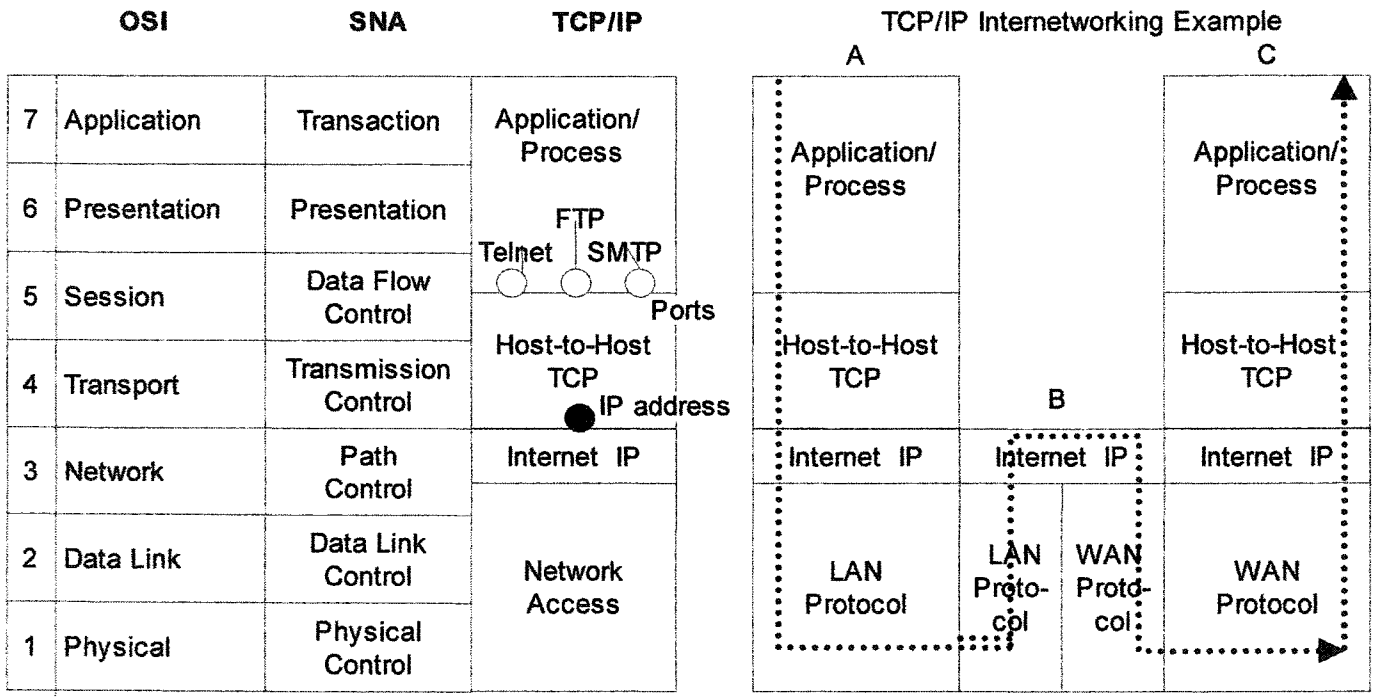
4.1 Industry Standards

The role of protocols is crucial in communications between customers (as FIIN service subscribers) and FIIN, and also between FIIN and the carriers. The conformance of SS7 to OSI model [12] is shown on Figure 73. Comparison of OSI, SNA, and TCP/IP [14, 15] along with TCP/IP internetworking example [13, 16] depicted in Figure 74. FIIN can access Network using any of these protocols from any platform.



ASE = Application Service Elements; ISUP = ISDN User Part; MTP = Message Transfer Part;
NSP = Network Service Part; OMAP = Operations, Maintenance and Administration Part;
SCCP = Signaling Connection Control Part; TCAP = Transaction Capability Application Part;
FCS = Frame Check Sequence; A = Address field; C = Control field; F = Flag sequence;
XH = Header block at layer X, where X = A, P, S, T, N.

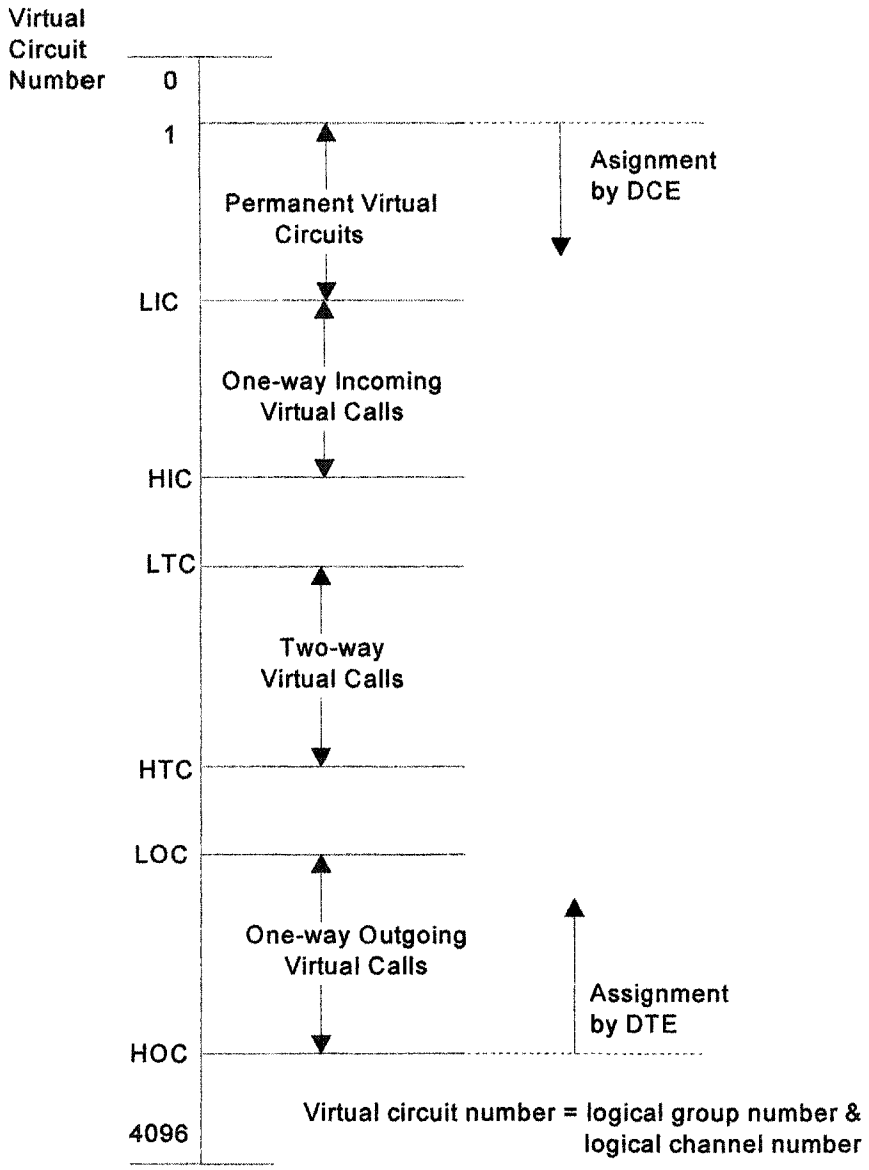
Figure 73 OSI and SS7 Protocols.



OSI = Open System Interconnection; SNA = System Network Architecture; TCP = Transmission Control Protocol; IP = Internet Protocol; LAN = Local Area Network; WAN = Wide Area Network; FTP = File Transfer Protocol; SMTP = Simple Mail Transfer Protocol; Telnet = Terminal Emulation Protocol.

Figure 74 Comparison of Communication Protocols.

FIIN connects to the carrier over shared , dedicated or lease line using X.25 protocol [18]. Assignment of virtual sequence numbers is shown on Figure 75 [**11**]. Two permanent virtual circuits normally used to establish a connection with the public carrier for incoming and outgoing messages (Figure 67).



PVC = Permanent Virtual Circuit; DCE = Data Circuit-Terminating Equipment; DTE = Data Terminal Equipment; LIC = Lowest Incoming Channel; HIC = Highest Incoming Channel; LTC = Lowest Two-way Channel; HTC = Highest Two-way Channel; LOC = Lowest Outgoing Channel; HOC = Highest Outgoing Channel.

Figure 75 Virtual Circuit Number Assignment.

Proprietary networks are usually compliant with standards organizations (ISO, ISITC).

4.2 Proprietary Protocols with Private Carriers

Application layer protocols between FIIN and SWIFT (Figure 76 and Figure 77) for outgoing and incoming messages correspondingly can be used as a base to develop international standards.

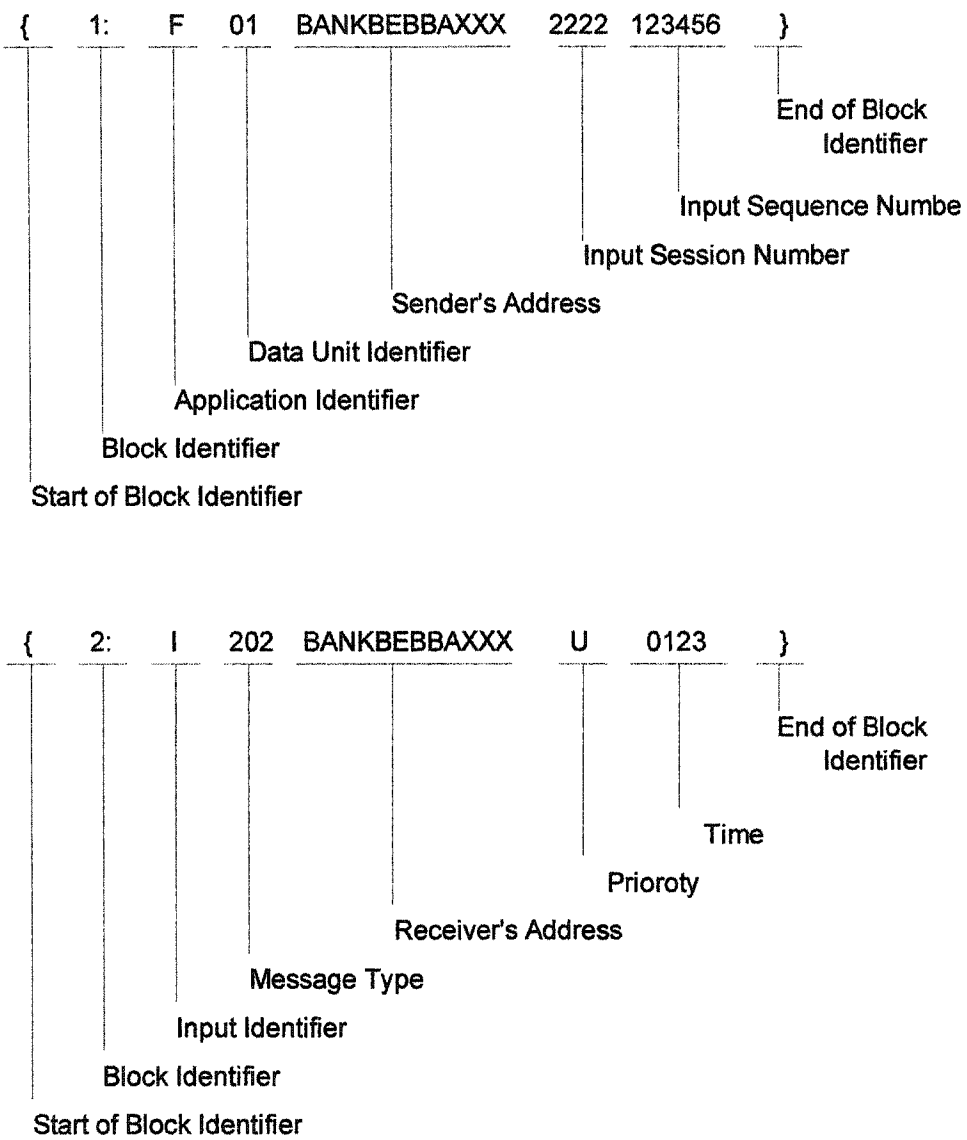


Figure 76 Application Header Format for Input to Network in the Protocol Between FIIN and SWIFT.

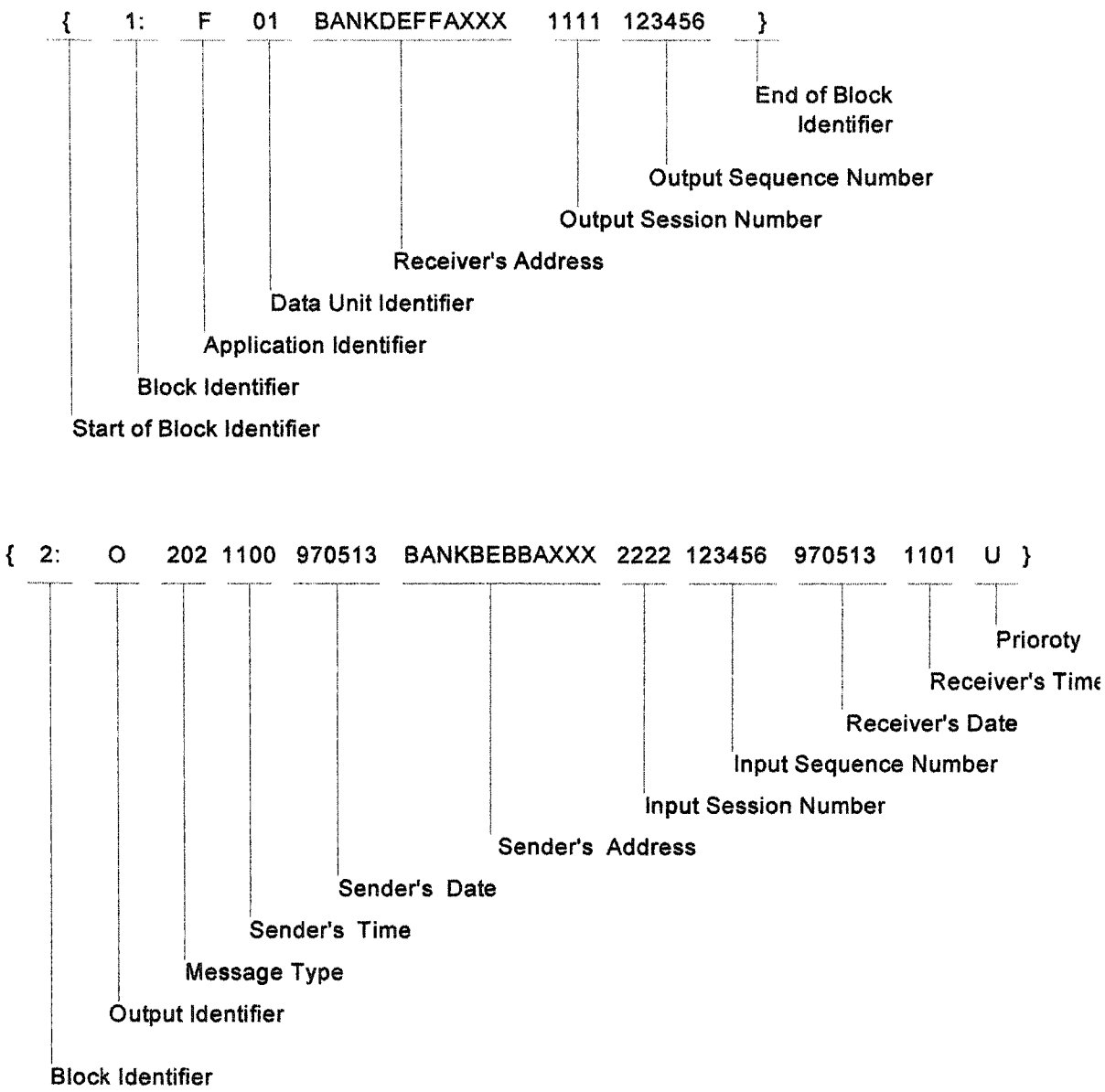


Figure 77 Application Header Format For Output from Network in the Protocol Between FIIN and SWIFT.

4.3 Proprietary Protocols with Public Carriers

Application layer protocols between FIIN and IC are shown on Figure 78 and Figure 79.

D	Ref	Route	Prio	Dup	Address	Text
I	Ref	Type	Trans Time			
S	CRef					

(a) Messages Generated by the Sender (FN).

Ref = Reference number assigned by FN;
CRef =Reference number assigned by IC;
Route=Route indicator: T = Telex,
C = Cable,
F = Fax;
Prio = Priority is the time IC tries to deliver Telex message before converting it to cable or cancelling it;
Dup = Possible Duplicate;
Address= Telex address and answerback cable address, fax number; one or more for each;
Trans Time= Time at which the message was transmitted from FN to IC.

S	Ref							
H	Ref	RS						
HOFF								
Q	CRef	Ref	RS					
C	CRef	Ref	T	Trans Time	Elapsed Time	Answerback	Charge	
C	CRef	Ref	C	Conv Time	Word Count		Charge	
C	CRef	Ref	F	Trans Time	Elapsed Time		Charge	
N	CRef	Ref	RS	Optional Text				
R	CRef	Ref	W	Retry Count				
R	CRef	Ref	T	Trans Time	Elapsed Time		Charge	Text
R	CRef	Ref	T	Conv Time	Word Count		Charge	Text
R	CRef	Ref	N	Reason	Optional Text			
R	CRef	Ref	U					
R	CRef	Ref	I					

RS = Reason for retransmission or reject;
T = Telex confirmation; C = Cable confirmation; F = Fax confirmation;
W = Waiting delivery; N = Rejected; U = Unknown format;
I = Inquiry failed.

(b) Messages Generated by the Carrier (IC).

FN = Financial Network; IC = Interexchange Carrier; D = Data Message; I = Inquiry;
S = Synchronyzation;H = Halt; HOFF = Resume; Q = Retransmit; C = Confirmation;
N = Rejection; R = Response to Inquiry.

Figure 78 Application Header Format for Input to Network in the Protocol Between FIIN and IC.

D	Ref	CRef	Dup		Answerback	Dialed No	Text
X	Ref	CRef	Dup	ORef	Answerback	Dialed No	Text

(a) Messages Received From the Carrier (IC) Generated by the Sender (FN).

Ref = Reference number assigned by IC;
CRef =Reference number assigned by IC for recovery;
ORef =Reference number of the original message assigned by Sender;
Dup = Possible duplicate;
Dialed No = Number dialed by Sender;
RS = Reason to halt transmission.

A	Ref	CRef	
H	Ref	CRef	RS
R	Ref	CRef	

(b) Messages Generated by the Receiver (FN).

FN = Financial Network; IC = Interexchange Carrier;
D = Data Message; X = Retrieved Data Message;
A = Acknowledgement; H = Halt Transmission; R = Resume.

Figure 79 Application Header Format for Output from Network in the Protocol Between FIIN and IC.

5 Functionality of Financial Institution Intelligent Network

5.1 Sending and Receiving Financial Messages

5.1.1 Data Entry and Graphic User Interface

5.1.1.1 Features of Data Entry and Verification (DEV)

Description of DEV package is given for a MERVA-based messaging system utilizing MERVA commands processed by developed APIs. MERVA is an IBM product, APIs were developed in this work.

Command	**Function**
DSLE	To Start DEV
RETURN **xxxDEVEI**	To Enter a Message
RETURN **xxxDEVVI**	To Verify a Message
RETURN **xxxDEVJI**	To Repair a Rejected Message
RETURN **xxxDEVBI**	To Inquire about a Specific Message.
RETURN **xxxDEVTI**	To Create a New or Modify an Existing Template of a Message

PF KEYS Description:

PF KEY	Various Functions Assigned
PF-1	HELP
PF-2	RETRIEVE last command
PF-3	• RETURN • EOM (End Of Message) • Signoff
PF-4	• Get MSG - in xxxDEVVI, bring cursor to the left of a message, press PF-4 to select it for verification. • TRACE - displays the trace for a message, including userids of users that data entered the message.
PF-5	Toggle Edit Display. Applicable to the free format messages with "Edit" switch set to "Y"
PF-6	REQUEUE currently viewed message. Place it at the end of the queue. While in xxxDEVEI/xxxDEVJI, save the message as a draft, for later retrieval.
PF-7	PREVIOUS Page (Page -1), Page BackWard
PF-8	NEXT Page (Page +1), Page ForWard
PF-9	PRINT currently viewed message (or screen) on a dedicated printer.
PF-10	• LIST the LAST page of messages, on the message selection screen. • Split the edited line in the middle and Insert a empty new line (for tag 79 only).
PF-11	LIST the FIRST page of messages on the message selection screen.
PF-12	• ESCAPE - discard currently edited message. • Go back to the prior menu. • LIST OFF.

5.1.1.2 DEV Message Flow

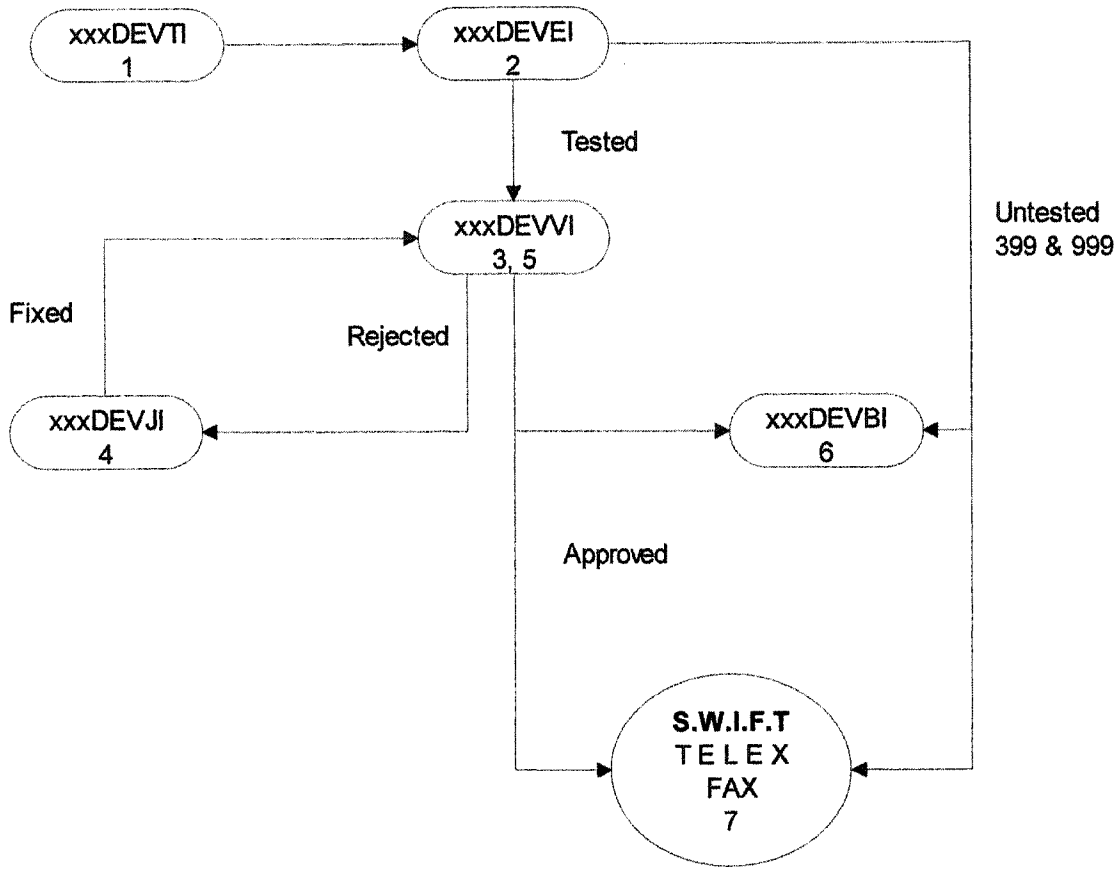


Figure 80 DEV Message Flow.

Description:

1. Template messages are created and stored on "xxxDEVTI" queue where xxx is country and city codes such as xxx={FRP, US3, BEB, ...}.
2. A template is copied from "xxxDEVTI" queue, or a new message is created in "xxxDEVEI" queue, and routed to "xxxDEVVI" for verification.

3. A supervisor reviews the message in "xxxDEVVI" queue and either approves or rejects the message.
4. If the message is approved, it is sent out to SWIFT. A copy of the processed message is kept on "xxxDEVBI" queue.
5. If the message is rejected, it is routed to "xxxDEVJI" queue for repair.
6. The person who originally data entered this message, repairs it on "xxxDEVJI" queue, and releases it back to "xxxDEVVI" for yet another verification.
7. The supervisor verifies and approves the repaired message, and then releases it to SWIFT.
8. The successfully processed message is kept on "xxxDEVBI" queue during retention period (usually 30 business days).
9. Supervisor has to verify and approve the message, in order to release it to SWIFT.

The facilities available to carry out the following tasks:

- Data Entry
- Verification
- Repair
- Inquiry
- FORMS – Templates for frequently used forms.

All users have to have access to the following:

- CICS Terminal or Application owning region (TOR or AOR)
- DSLE Main CICS transaction for running DEV in MERVA.
- Functions:

  xxxDEVEI,

  xxxDEVJI
- Authorized personnel should also have access to functions:

  xxxDEVVI,

  xxxDEVBI,

  xxxDEVTI.

See more detailed description in Appendix A.

5.1.2 Internal Routing

FIIN in internal software defined network can simulate external network. FIIN delivers messages between its internal branches or acting on behalf of other directly linked customers as the service bureau. I-type message are converted to O-type calculating and verifying the trailers and delivering messages too proper back-end application.

5.1.3 Application Interfaces

Application interfaces are the following:

- MQ-based,

- FTP-based,
- Proprietary, Data entry GUI inside BMS.

5.1.4 Charging, Statistics and Archiving

Message charging depends on networks tariffs and can vary during the day. Therefore current charge per message can be a parameter in BMS for making a routing decision.

Statistics has to have variable profile, collected instantly and provided on request in WEB-browsable format.

Messages are archived for 7 years and available online with the audit trail for investigation.

5.2 Service Applications

5.2.1 Software Defined Internal Network

The table of all FIs connected to Intranet is maintained and easily updated. FIs have to be directly linked via the head company Intranet. All security arrangements have to be made.

5.2.2 Message Scanning

In the queuing messaging system each message proceeds through the number of queues (hops) between a back-end application and a network. In the developed BMS this number is about 15 on each way that is quiet expensive in

sense of I/Os and performance. The contemporary approach is to keep all intermediate hops in memory preserving commit points on first and last queue in case of failure and gaining substantially in performance. This is called message scanning or I/O interpreting.

5.2.3 OFAC Scanning

The Office of Foreign Assets Control of the U.S. Department of the Treasury enforces economic and trade sanctions against targeted foreign countries, terrorism sponsoring organizations and international narcotics traffickers based on U.S. foreign policy and national security goals. OFAC acts under Presidential wartime and national emergency powers, as well as authority granted by specific legislation, to impose controls on transactions and freeze foreign assets under U.S. jurisdiction. Many of the sanctions are based on United Nations and other international mandates, are multilateral in scope, and involve close cooperation with allied governments (see http://www.treas.gov/ofac).

The example of OFAC scanning of messages by BMS is shown on Figure 81 and Figure 82.

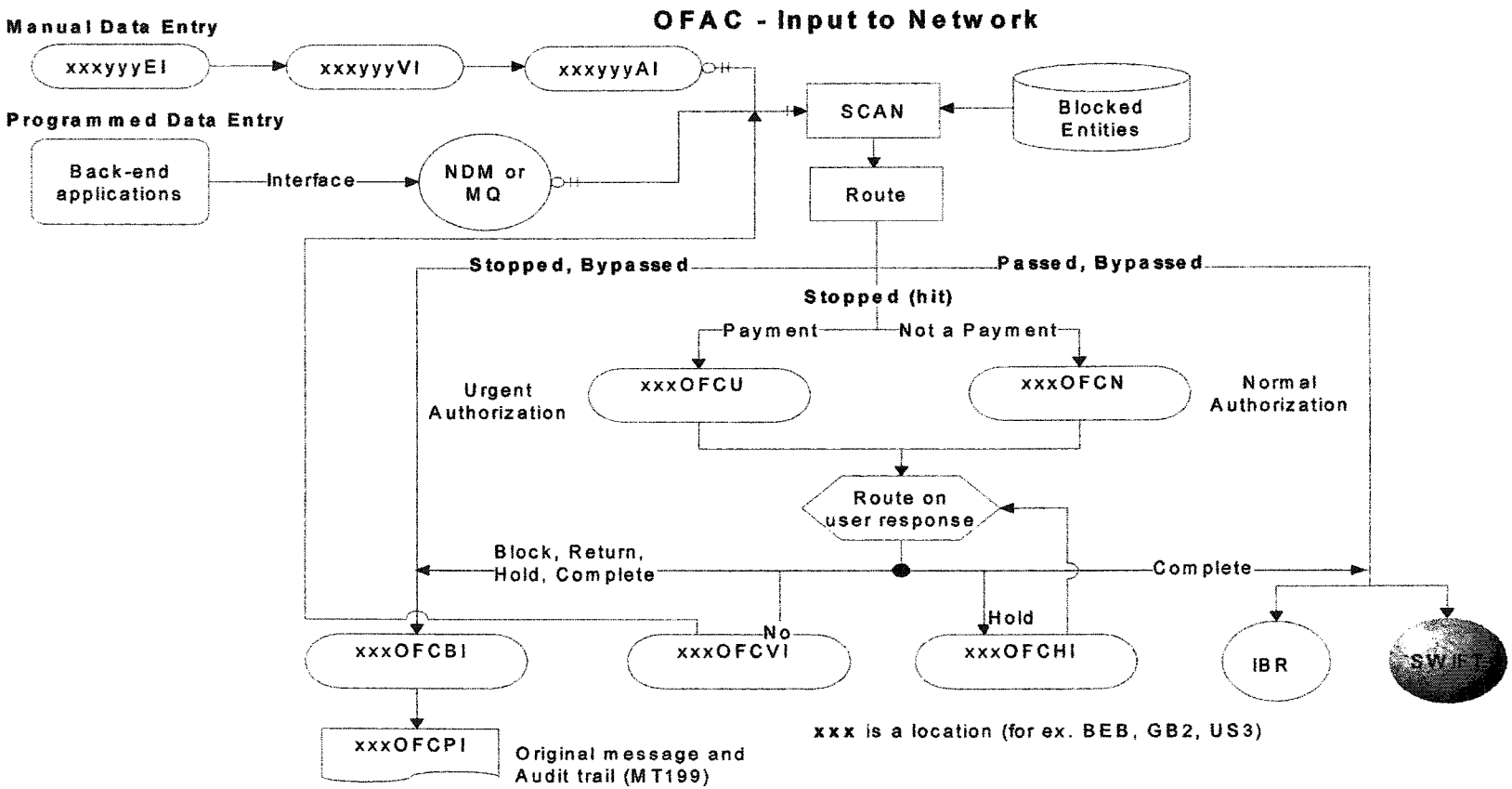


Figure 81 OFAC – Input to Network..

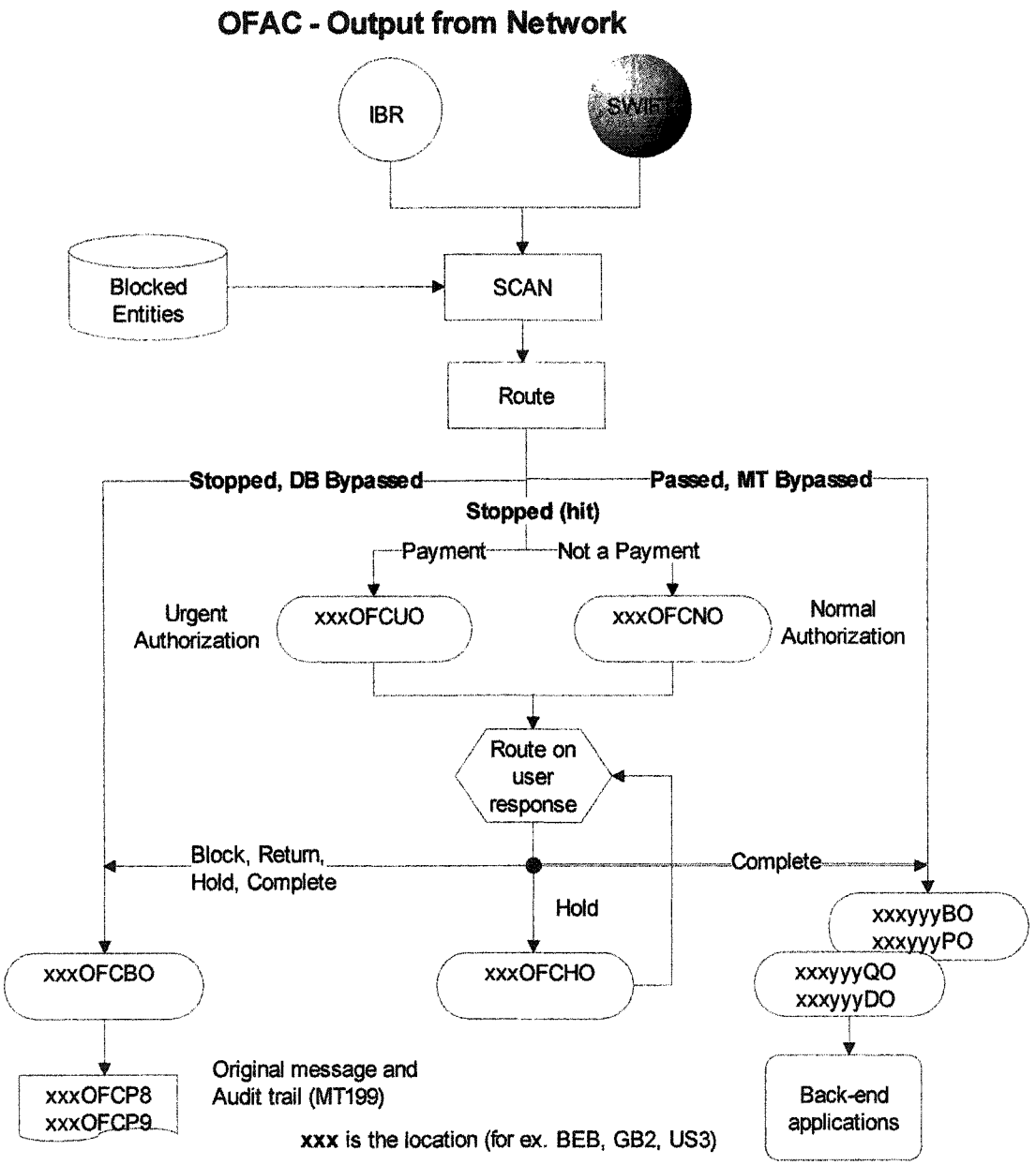


Figure 82 OFAC – Output from Network.

5.2.4 Message Reconciliation and Matching

Each message is processed through BMS in three basic steps:

O – Original,

A – acknowledged (I-type) or application identified (O-type),

F – Final.

Message is reconciled if it passes all 3 steps. It means that I-type message sent to network, ACK is received by back-end application and the message is archived. O-type message is received from network, ACK is received by network, the message delivery is confirmed by back-end application and the message is archived in BMS.

5.2.5 Message Re-advising

Message received by recipient A from B can be re-advised to party C for some political reasons when A plays the mediator's role. BMS automate this process by manipulating O-type message into I-type with pre-agreed modifications.

5.2.6 Continuous Linked Settlement (CLS)

The closed group of banks are involved in so-called Continuous Linked Settlement to reduce the associated risk (see Figure 83 and Figure 84).

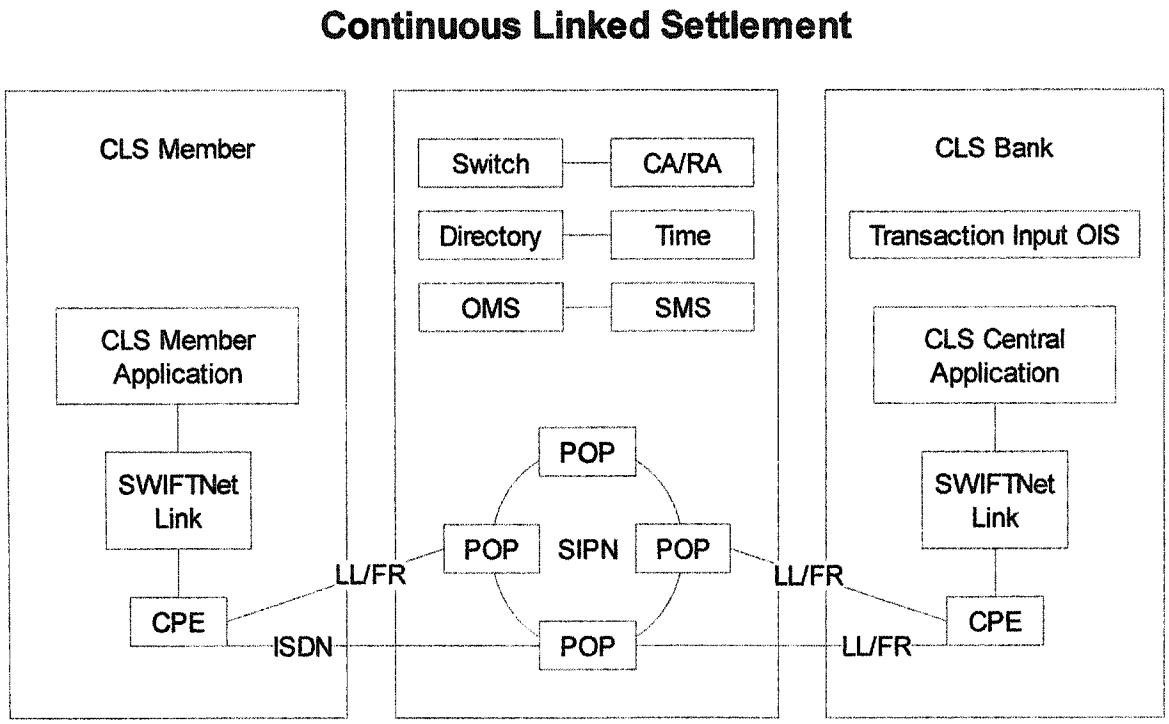


Figure 83 Continuous Linked Settlement.

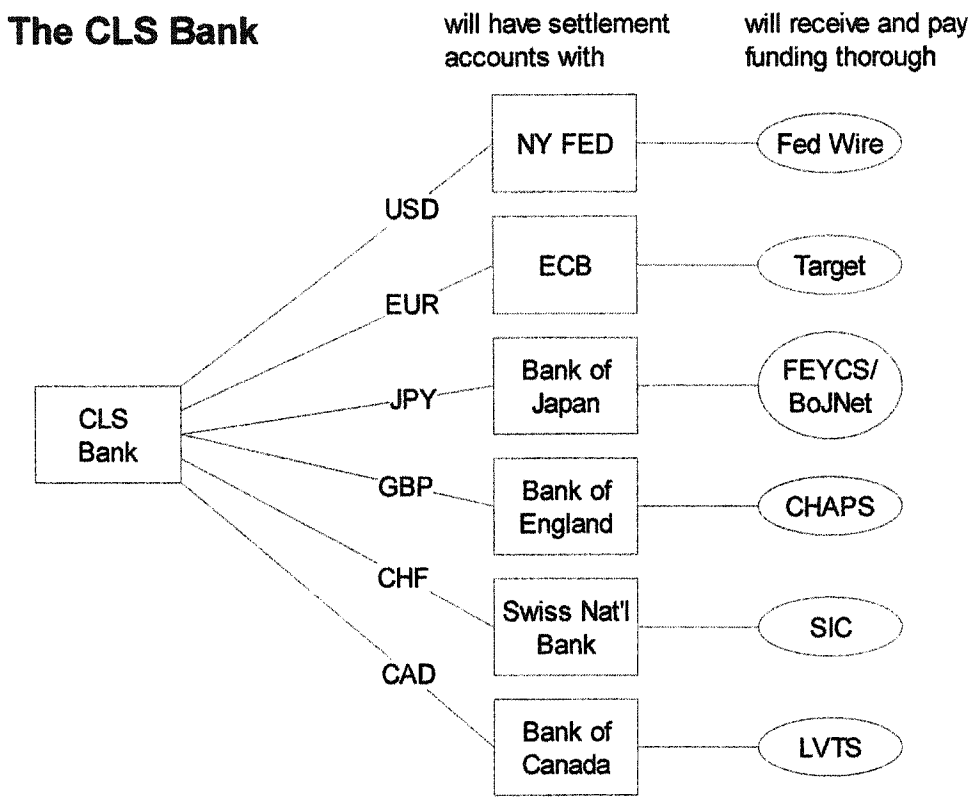


Figure 84 The CLS Bank.

6 Conclusion

6.1 Contribution

Use of Advanced Intelligent Networks (AIN) undergoes considerable growth as the result of fast advances in network technologies. Application of Intelligent Peripherals (IP) is rapidly changing along with AINs. Corporations can use IP approach to build the entire network customized to their specific needs. Such an IP-network is integrated into AIN current implementation (as Public Switched Network) utilizing existing signaling system.

The described original method of constructing IP is offered for financial institutions but can be used by any large corporation with varied lines of business (insurance, medical, knowledge-based, telemarketing, etc.).

Contribution is made in the area of network architecture and design. Presented new technique, concept and analysis of Financial Institution Intelligent Network (FIIN) constructed as Intelligent Peripheral (IP) with entire Intelligent Network (IN) architecture specifically designed for routing and transmission of financial messages in context of Advanced Intelligent Network (AIN). FIIN can serve local or wide geographically dispersed area connecting FI intranet with partner FI intranet, national network (like FedWire, CHIPS) or worldwide network (SWIFT, Internet, Telex/FAX carriers).

Detailed functional specifications are given for FIIN building blocks: Message Switching Point (MSP), Message Transfer Point (MTP), Message Control Point (MCP). Described Message Control System (MCS) monitoring message flow within FIIN and between FIIN and AIN components. Shown protocols between MCS and SS7, links between MTP and Service Transfer Point (STP) of AIN.

Contribution in the area of communication protocols includes the following:

- design of four sub-layers of OSI application layer: application interface, application processing, message processing and network selection;
- examples of proprietary protocols with public carriers;
- utilization of existing protocols, protocol and format conversion.

Protocol design allows for customized message routing, congestion avoidance, flow control and prioritization on any sub-layer level, network selection and access.

Contribution in network software, services and applications includes connection-oriented services with location-independent communications, application-driven financial messaging systems with predefined business rules. Developed software packages spread out over multiplatform environments (IBM Mainframe, VAX, UNIX, PC). Utilized existing software on communication controllers (IBM 3745) and routers (CISCO) for varied file/message transfer mechanisms with underlying TCP/IP or X.25 protocols.

Developed network signaling software separating business data from different levels of acknowledgments (ACK/NAK), confirmations (COA, COD), rejections, delivery notifications, system and administrative messages, removing error messages for repair. Location-specific processing is separated and independent from application-specific. New SCE language for FIIN facilitates location-application combinations as set of macros and standards (naming conventions) for message queues. End-user customizations do not depend on application modifications and enhancements.

Developed operation, administration and management software for FIIN that provides network monitoring, special routing for network security (login and bilateral keys), handles libraries, data bases, statistics, charging, service creation, version upgrades and related functions. Monitoring procedures for standardized message queues provide scalable solution for increasing traffic and throughput, prioritization mechanism, troubleshooting, improved error correction and intelligent message flow distribution.

FIIN is successfully implemented for a major bank. Described functionality of the real FIIN system with value added services.

6.2 Prospective

More cooperation has to be developed between FIN and AIN. Development and implementation of MCS will allow direct communication

with SS7 between MTP of FIN and STP of AIN. Protocols will be worked out, formalized, and standardized.

Work on construction of three major FIN building blocks will continue to precisely define and implement each one of them: MSP, MTP, and MCP. They will be offered by FIN vendors.

SMS and SCE will be improved for FIN to introduce new banks and brokerage applications for transaction-message processing.

Intelligent portion of FIN will be modified and enhanced to make FIN more flexible, more secured and easy to implement.

7 Appendix

7.1 Appendix A

7.1.1 Data Entry Function

Function xxxDEVEI:

```
Message Selection                                        Func xxxDEVEI

    To Create a New Message (Message Type) Enter:

       Command =====> MT Snnn


    To Copy a Message from xxxDEVTI Queue by TRN Enter:

       Command =====> COPY K2 'YYYYYYYYYZZZZZZZZZZZZZZZ'



Command =====> 399
 PF 1=Help 2=   3=Return     4=          5=              6=
 PF 7=     8=   9=          10=         11=List First   12=
```

New message can be data entered or copied from a template.

Optional Fields:

"Test? Y/N" Applicable only to message types n99. All other messages are tested (option N is not available for other messages).

"Test amounts? Y/N" If a "Y" is entered, pressing PF8 brings up the next screen.

"Third Party Charging" is an optional (usually 10-digit) numeric field that can be used to charge a third party. User can enter a Cost Center number or an Account number of a third party. If left blank, the default is the Cost Center number of the Sender of the message.

```
S399                          Free Format                    Page 00001
                                                             Func xxxDEVEI

Sender code:_________                  Test? Y/N:__
                                       Test amounts? Y/N: __
Recipient                              Third Party Charging: _________
      ID Type :_______          (CID, NMK, SWF, ABA, CHP, ACT, NCF)
      ID Value
To Name:
Addr -1:
Addr -2:
Country:
------------------------------------------------------------------------
Message                    : S399 Initialized
------------------------------------------------------------------------
Basic Header          F  01 ABCDBEB0AXXX 0000 000000
Application Header    I 399              N
User Header           Service Code_____     103:_____
                      Bank. Priority  113:______
                      Msg User Ref.   108:______


Command ======>
PF 1=Help      3=EOM       4=Trace  5=Toggle Edit Display    6=Requeue
PF 7=Page Bkw  8=Page Fwd  9=Hardcopy  10=Insert  11=       12=Escape
```

Application Header is automatically generated based on CID, NMK, SWF, ABA, CHP, or ACT. Any user entered input in this field is replaced.

TRN *20: is also generated automatically

Mnemonic	Description	Number of Chars	Example
ABA	American Banking Association Number	9	012300028
ACT	Bank Account Number	10	8012345678
CHP	New York Clearing House Member number	8	00123456

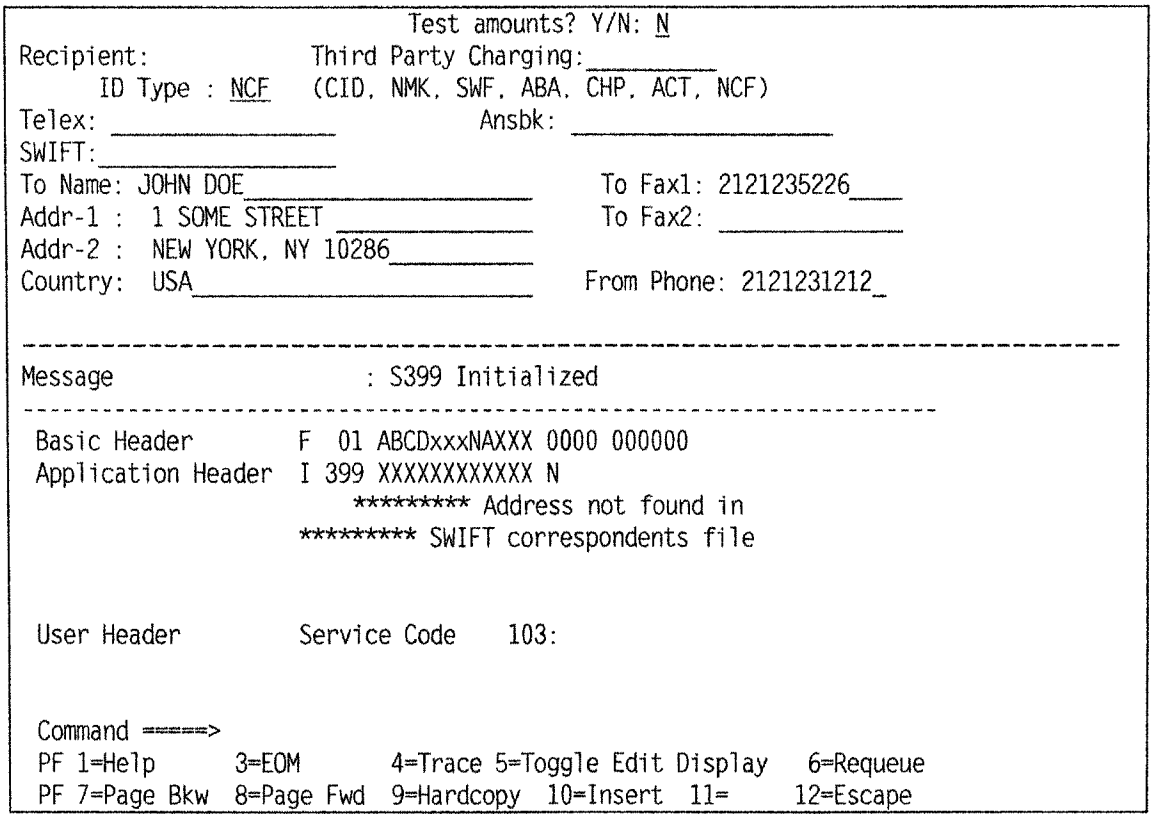
CID	CIF Customer ID	10	9123456789
NMK	CIF Name Key with Tiebreaker	15	CREDLIBAB EI3001
SWF	SWIFT Address	11-max, 8-min	CLIBLBBXB RN
NCF	NOT ON CIF	N/A	N/A

```
MT S399                        Free Format                      Page 00001
                                                              Func xxxDEVEI

Sender code: 1234567AB        Test? Y/N: Y
                              Test amounts? Y/N
Recipient:                 Third Party Charging:_____
      ID Type : CID    (CID, NMK, SWF, ABA, CHP, ACT, NCF)
      ID Value:   1234123567
To Name: BANK ABCD
Addr-1:  INTERNATIONAL DIVISION
Addr-2:  POB 12348, CHURCH ST STATION
Country: NEW YORK, NY 10286
--------------------------------------------------------------------------------
Message                     : S399 Initialized
--------------------------------------------------------------------
 Basic Header         F  01 ABCDxxxNAXXX 0000 000000
 Application Header   I 399 ABCDLULXAXXX N
                            *BANK OF CITY
                            *NEW YORK,NY
 User Header          Service Code    103:


 Command =====>
 PF 1=Help      3=EOM       4=Trace 5=Toggle Edit Display   6=Requeue
 PF 7=Page Bkw  8=Page Fwd  9=Hardcopy  10=Insert  11=     12=Escape
```

When "Test" switch is set to "N" message types n99 are sent untested. It means that these messages do not require verification and send directly to their destinations. All other messages are tested requiring verification by a different user (function xxxDEVVI) due to 4-eye principle.

Message also can be Faxed:

```
MT S399                        Free Format                      Page 00001
                                                              Func xxxDEVEI

Sender code: 1234567AB        Test? Y/N: N
```



Payment message has to be verified by re-typing the amount information. Message can be rejected or repaired to reenter the message flow. The broadcast message can be sent using the distribution list.

The DEV Broadcast Message can be sent Tested or Untested. If a message is tested, it is routed to the Verification Queue after all the required fields have been entered. After verification the message is routed to an intermediate queue where it is replicated and one copy is routed to each Recipient. If a message is untested, it is routed to the intermediate queue after all the required fields have been entered. Untested message is also replicated and routed from the intermediate queue to each Recipient.

7.1.2 Special DEV Features

7.1.2.1 Using xxxDEVTI Queue

"xxxDEVTI" function is used to create and maintain templates of frequently used messages. The template messages can later be copied into "xxxDEVEI" for quick and easy preparation of standard free-formatted messages.

Preparing a Template Message in xxxDEVTI:

```
                    Message Selection                    Func xxxDEVTI


        To Create a New Message (Message Type)       Enter:

Command ===>  MT Snnn


Command =====> 199
PF 1=Help 2=   3=Return     4=          5=              6=
 PF 7=     8=   9=          10=         11=List First   12=
```

To create a new template select "xxxDEVTI" from MERVA DEV Function Selection Screen. The above screen is displayed as a result.

User Response:

- Enter "MT" followed by a blank, followed by an "S", and followed by a 3-digit Message Type number. Alternatively simply enter the 3-digit message type.
- Press "ENTER".

Creating a Template Message in xxxDEVTI:

```
S199                          Free Format                    Page 00001
                                                             Func xxxDEVEI

Sender code: 1234567AB                Test? Y/N:__
                                      Test amounts? Y/N: __
Recipient                             Third Party Charging: _______
      ID Type :_______          (CID, NMK, SWF, ABA, CHP, ACT, NCF)
ID VALUE:
To Name:
Addr-1:
Addr2:
Country:
--------------------------------------------------------------------------------
Message                    : S399 Initialized
-------------------------------------------------------------------
 Basic Header        F  01 ABCDxxxNAXXX 0000 000000
 Application Header  I 399 XXXXXXXXXXXX N
                         ********* Address not found in
                       ********* SWIFT correspondents file


 User Header         Service Code    103:


 Command =====>
 PF 1=Help      3=EOM       4=Trace 5=Toggle Edit Display   6=Requeue
 PF 7=Page Bkw  8=Page Fwd  9=Hardcopy  10=Insert  11=     12=Escape
```

After typing "MT S199" on the command line and pressing "ENTER" the above screen is displayed.

User Response:

- Enter mandatory "Sender Code:".
- Enter unique "TRN". This TRN is used to identify the template. In this example "MY-199" is used as a unique TRN.
- Follow by any other generic information that can be reused.
- Press PF-8 to enter generic Free Text.

Creating a Template Message in xxxDEVTI, Page-2:

```
MT S199                          Free Format                          Page 00002
                                                                    Func xxxDEVTI
Bank. Priority  113:
Msg User Ref.   108:
TRN             *20: MY-199
Related Reference    21:
Narrative           *79: WHEN CREATING THE TEMPLATE
TAG-20 IS NOT GENERATED AUTOMATICALLY.
ALSO, RECIPIENT INFORMATION IS NOT VALIDATED.
MERVA DEV LETS YOU SAVE THIS MESSAGE IN AN
INCOMPLETE FORMAT AS A TEMPLATE.

Command =====>
 PF 1=Help      3=EOM       4=Trace      5=Toggle Edit Display    6=Requeue
 PF 7=Page Bkw  8=Page Fwd  9=Hardcopy  10=Insert    11=         12=Escape
```

User Response:

- Data enter generic free text that can be reused later.
- Press PF-3 to store the template on "xxxDEVTI" queue.

Storing a Template Message in xxxDEVTI:

```
                                Message Selection                    Func xxxDEVTI

To Create a New Message (Message Type)       Enter:



Command ===>  MT Snnn

DEV017E MESSAGE SUCCESSFULLY STORED
Command =====> MT S199
 PF 1=Help      2=          3=Return     4=          5=              6=
 PF 7=          8=          9=          10=         11=List First   12=
```

Copying a Template Message from xxxDEVTI:

```
                          Message Selection                    Func xxxDEVEI



To Create a New Message (Message Type)       Enter:

Command ===>  MT Snnn


   To Copy a Message from xxxDEVTI Queue by TRN Enter:
COPY K2 'yyyyyyyyyyzzzzzzzzzzzzzzz'


Command ======> COPY K2  '1234567ABMY-199'
 PF 1=Help      2=          3=Return     4=          5=              6=
 PF 7=          8=          9=           10=         11=List First   12=
```

User Response to COPY a template:

- On the command line type: "COPY K2 '1234567ABMY-199' " (in this example, '1234567AB' is a Sender Code; "MY-199" is a TRN that uniquely identifies the template to copy).
- Press "ENTER".

Copying a Template Message from xxxDEVTI, Page-1:

```
 MT S199                        Free Format                    Page 00001
                                                               Func xxxDEVEI

 Sender code: 1234567AB   Test amounts? Y/N:_

 Recipient                Third Party Charging: _________

ID Type :       (CID, NMK, SWF, ABA, CHP, ACT, NCF)
ID Value:
To Name:
Addr-1:
Addr-2:
Country:
--------------------------------------------------------------------------------
Message                   : S199 Initialized
--------------------------------------------------------------------------------
 Basic Header        F  01 ABCDxxxNAXXX 0000 000000
 Application Header  I 199 XXXXXXXXXXXX N
```

```
                    ********* Address not found in
                    ********* SWIFT correspondents file

User Header         Service Code    103:


Command ======>
PF 1=Help      3=EOM       4=Trace 5=Toggle Edit Display   6=Requeue
PF 7=Page Bkw  8=Page Fwd  9=Hardcopy  10=Insert  11=     12=Escape
```

User Response:

- Fill in all the required fields.
- Press PF-8, to enter text on the next page.

Copying a Template Message from xxxDEVTI, Page-2:

```
MT S199                      Free Format                        Page 00002
                                                                Func xxxDEVEI
Bank Priority 113:
Msg user ref  108:
TRN                 *20    : AMV9907050056800
Related Reference    21    :
Narrative           *79    : WHEN CREATING THE TEMPLATE
TAG-20 IS NOT GENERATED AUTOMATICALLY.
ALSO, RECIPIENT INFORMATION IS NOT VALIDATED.
    MERVA DEV LETS YOU SAVE THIS MESSAGE IN AN
    INCOMPLETE FORMAT AS A TEMPLATE.



Command ======>
 PF 1=Help      3=EOM       4=Trace      5=Toggle Edit Display    6=Requeue
 PF 7=Page Bkw  8=Page Fwd  9=Hardcopy  10=Insert     11=         12=Escape
```

User Response:

- Retain the automatically generated TRN Information identified by "TRN *20 :". This Transaction Reference is used to uniquely identify the message. It replaces "MY-199" that identified the template.
- Data enter free text in the space provided next to the mandatory tag 79.

- Press either one of the following PF keys:

  PF-3: To end and release the message. (EOM - **E**nd **O**ff **M**essage),

  PF-6: To save the message as a draft for later retrieval,

  PF-12: To abandon and discard the edited message.

7.2 Appendix B

7.2.1 Retrieving a Draft Message into Data Entry Queue

Any message can be saved as a DRAFT on xxxDEVEI queues. Later the saved draft can be retrieved. Following is retrieval example of a message saved as a draft on xxxDEVEI queue.

Selecting a Draft Message to Retrieve:

```
                          Message Selection                  Func xxxDEVEI


        To Create a New Message (Message Type)       Enter:

Command ===>  MT Snnn



    To Copy a Message from xxxDEVTI Queue by TRN Enter:
    COPY K2 'yyyyyyyyyzzzzzzzzzzzzzzz'


Command =====>
 PF 1=Help      2=          3=Return     4=          5=              6=
 PF 7=          8=          9=           10=         11=List First   12=
```

User Response:

- Press PF-11 to display the list of messages waiting to be processed on the data entry queue. The first page of messages is displayed:

```
MT S199                          Free Format                          Page 00002
                                                                      Func xxxDEVEI

TRN

> AMV0306030045100
> AMV0302290040600
> AMV0306040045400
> AMV0306100046900
> AMV0306230052300
> AMV0307020055900
> ANV0306080045900

To select a message, move cursor to ">" and press PF4
Select by SENDER Code: 1234567AB                  TRN


Command ======>
PF 1=Help       2=Retrieve  3=Return     4=Get MSG    5=            6=
PF 7=List Back 8=List Fwd  9=Hardcopy  10=List Last 11=List First 12=List Off
```

User Response:

- Move cursor to the message to be edited. The message can be identified by TRN.
- Press PF-4 to select the message for editing.

Retrieving a Draft Message:

```
 MT S199                          Free Format                          Page 00001
                                                                       Func xxxDEVEI

 Sender code: 1234567AB   Test amounts? Y/N:_

 Recipient                Third Party Charging: _________

ID Type : CID     (CID, NMK, SWF, ABA, CHP, ACT, NCF)
ID Value: 9123123879
To Name:  Credit Bank
Addr-1:   Ocean Blvd
Addr-2:   Investor Center
Country:   City, Country
--------------------------------------------------------------------------------
Message                  : S199 Initialized
--------------------------------------------------------------------------------
 Basic Header        F  01 ABCDxxxNAXXX 0000 000000
 Application Header  I 199 ABCDUS3NAXXX N
```

```
                        *Bank of Sample
                        *New York, NY

User Header          Service Code    103:


Command ======>
PF 1=Help      3=EOM       4=Trace 5=Toggle Edit Display   6=Requeue
PF 7=Page Bkw  8=Page Fwd  9=Hardcopy  10=Insert  11=     12=Escape
```

User Response:

- Review, correct or complete the necessary information.
- Press PF-3 to complete and release this message or
- Press PF-8 to go to the next page.

Retrieving a Draft Message, Page-2:

```
MT S199                      Free Format                      Page 00002
                                                              Func xxxDEVEI
Bank Priority 113:
Msg user ref  108:
TRN                  *20   : AMV0306080045900
Related Reference     21   :
Narrative            *79   : MESSAGE WAS REQUEUED WITH PF-6
NOW IT CAN BE RETRIEVED.


Command ======>
 PF 1=Help      3=EOM       4=Trace      5=Toggle Edit Display    6=Requeue
 PF 7=Page Bkw  8=Page Fwd  9=Hardcopy  10=Insert    11=         12=Escape
```

User Response:

- Enter text for tag-79.
- Press PF-3 to complete and release this message to verification.

Completing and Releasing Retrieved Draft Message:

```
MT S199                          Free Format                          Page 00002
                                                                      Func xxxDEVEI
TRN
> AMV0306030045100
> AMV0302290040600
> AMV9906040045400
> AMV9906100046900
> AMV9906230052300
> AMV9907020055900


To select a message, move cursor to ">" and press PF4
Select by SENDER Code: 1234567AB                    TRN

DEV001I MESSAGE IS ROUTED TO xxxDEVVI
Command ======>
PF 1=Help      2=Retrieve  3=Return     4=Get MSG    5=            6=
PF 7=List Back 8=List Fwd  9=Hardcopy  10=List Last 11=List First 12=List Off
```

User Response:

- Press PF-3 to go back to the main menu.

Bibliography

1. S.V. Ahamed, V.B. Lawrence. 1997. "Intelligent Broadbank Multimedia Networks." Kluwer Academic Publishers. Boston, MA.

2. S.V. Ahamed, V.B. Lawrence. 1997. "Design and Engineering of Intelligent Communication Systems." Kluwer Academic Publishers. Boston, MA.

3. S.V. Ahamed, M.J. Miller. 1988. "Digital Transmission Systems and Networks. Vol 1: Principles." Computer Science Press. Rockville. MD.

4. S.V. Ahamed, M.J. Miller. 1988. "Digital Transmission Systems and Networks. Vol 2: Applications." Computer Science Press. Rockville. MD.

5. S.V. Ahamed. 1989. "Chapter 9. Intelligent Networks" in Encyclopedia of Telecommunications. Academic Press (January): 159-174.

6. S.V. Ahamed, V.B. Lawrence. 1992. "Intelligent Networks: Architecture and Implications."Encyclopedia of Physical Sciences and Technology, Vol 8, Academic Press: 229-262.

7. J. Thorner. 1994. "Intelligent Networks." Artech House. Boston.

8. J. Martin. 1988. "Principles of Data Communications." Prentice Hall, Engelwood Cliffs, NJ.

9. U. Black. 1989. "Data Networks." Prentice Hall, Engelwood Cliffs, NJ.

10. W. Stallings, 1987. "Computer Organization and Architecture." Macmillian Publishing Company, NY.

11. W. Stallings, 1991. "Data and Computer Communications." Third Edition. Macmillian Publishing Company, NY.

12. W. Stallings, 1990. "Handbook of Computer Communications Standards. Vol 1: The OSI Model and OSI-related Standards." Second Edition. Macmillian Publishing Company, NY.

13. W. Stallings, 1990. "Handbook of Computer Communications Standards. Vol 2: Local Area Networks Standards." Second Edition. Howard Sams & Company.

14. W. Stallings, 1989. "Handbook of Computer Communications Standards. Vol 3: The TCP/IP Protocol Suit." Second Edition. Howard Sams & Company.

15. Networking and Systems Management Services and Support. 1992. "TCP/IP. Technical Overview and Coexistence." Washington System Center. Technical Bulletin. GG66-3238-00.

16. D. Chappell. 1997. Networking Series. Computer Channel Inc.

17. P. Houston. 1994. "Distributed Transaction Processing (DTP) Monitors." Enterprise Systems Journal, Dallas, TX.

18. IBM Corp. 1992. "X.25 Network Control Program. Packet Switching Interface." Research Triangle Park, NC. GC30-3469-05.

19. JenTech Inc. 1995. "Client/Server Application Development. Modern Technology Overview." Palm Bay, FL.

20. IBM Corp. 1993-1997. "Message Entry and Routing For Various Applications (MERVA)." Germany, Frankfurt. Industry Solutions Architecture. www.ibm.com/software/solutions/finance/merva/

21. Society For Worldwide Inerbank Financial Telecommunications (SWIFT). 1990-1997. Technical Manuals. Documentation Department. La Halpe. Belgium. www.Swift.com

22. Furash & Company. 1997. "Banking's Role in Tomorrow's Payments Systems." Vol 2. NY.

23. Raymond Panko. 1999. Business Data Communications and Networking. Second Edition. Prentice Hall. NJ.

24. F. S. Hillier, G. J. Lieberman. 2001. Introduction to Research. Seventh Edition. Mc-Grow-Hill.

25. Transaction Systems Architects, Inc. consists of ACI Worldwide, Insession Technologies, and IntraNet, Inc. www.TSAinc.com

26. SunGard Transaction Network, www.SunGard.com

27. Mercator Software Inc., www.Ascential.com

28. CHIPS Clearing House Interbank Payment System. http://www.chips.org/

29. FEDWIRE Network. http://www.ny.frb.org

30. NYCH New York Clearing House. http://www.nych.org/index.htm

Glossary

Item	Description
ABS	Alternate Billing Service
AIS	Application Interface Switch
ASCII	American Standard Code for Information Interchange
ASR	Application for Security Processing or Automatic Speech Recognition depending on the context
ATM	Asynchronous Transfer Mode
BIC	Bank Identification Code uniquely identifies FI. SWIFT is the registration authority.
BMS	Bank Messaging System
CA	Certificate Authority
Carrier system	System for modulating a periodic carrier signal with information of one or more channels to be able to transmit the combined signal over any specific transmission medium.
CCSS	Common Channel Signaling System (Q.931) provides the signal flow.
CD	Compact Disk
Channel	Logical connection between any two points in the network to exchange information. A network uses physical and logical addresses of a channel to convey information.
CHIPS	The Clearing House Inter-Bank Payments System is a real time final settlement payments system for B2B transactions.
CICS	Customer Information Control System
CIF	Customer Information File
Circuit switching	Mode of interconnecting logical channels between nodes of the network and releasing the channels after use to return to their idle state.
CLASS	Custom Local Area Signaling Services
CLS	Continuous Linked Settlement
CPE	Customer-Premises Equipment

DASD	Direct Access Storage Device
DEV	Data Entry and Verification
DLC	Directly Linked Customers
DVD	Digital Video Disk
EBCDIC	Extended Binary Coded Decimal Interchange Code
FDDI	Fiber Distributed Data Interface
FE ACK	Front-end ACK generated by BMS. Back-end application matches FE ACK with FRB ACK. FE ACK consists of ISN, IMAD, TRN, timestamp.
FEDWIRE	FRB electronic funds transfer (EFT) enables FIs transfer funds and securities within US.
FIFO	First-In First-Out
FIIN	Financial Institution Intelligent Network
FIN	Financial Institution Network
Financial infrastruc-tures	International Straight Through Processing Systems
Financial message	Single unit of work carried over financial network
Firmware intelligence	Microcoded into the control memories of the monitoring computers.
FIX	Financial Information Exchange is the securities' messaging standard
FLASH	Fed Link Access for Secondary Half-sessions is the message format to send/receive FedWire messages
FPI	Flesh Programming Interface to communicate with FedWire (supported by HP).
FR	Frame Relay
FRB	Federal Reserve Bank (also FED) is the central US bank, consists of 12 regional Reserve Banks.
FTP	File Transfer Protocol
FTS	Funds Transfer System is a FI back-end application.
IMAD	Input Message Accountability Data (Date, LT, FRB ISN)
IN	Intelligent networks are the carriers of information with distinct algorithmic adoption. Hardware intelligence resides in the customized integrated circuit chips, their sophisticated layout, and their interconnections.
Infomaster	Formerly Telex Computer Service (TCS) is a message switching service for Western Union Telex subscribers (offers Telex-to-TWX conversion).

IP	Intelligent Peripheral or Internet Protocol depending on the context
ISDN	Integrated Services Digital Network. Basic rate ISDN: 2B + D = 2 * 64 kbs + 1 * 16kbs; bi-directional 56kbs throughput capacity + 8kbs overhead.
ISO	International Standard Organization
LEC	Local Exchange Carrier
LT	Logical Terminal connecting to a Network
LL	Leased Line
LPAR	Logical Partition
LU	Logical Unit
MCP	Message Control Point
MCS	Message Control System
Message	Logical unit of data
MQ	Message Queue
MQSI	MQ Series Integration
MSP	Message Switching Point
MTE	Message Transformation Engine
MTP	Message Transfer Point or Message Transfer Part of SS7 depending on the context
NDM	Network Data Mover
NSP	Network Service Provider
NYCH	New York Clearing House (has over 1200 members supporting CHIPS)
OFAC	Office of Federal Asset Control
OMAD	Output Message Accountability Data (Date, LT, FRB OSN)
OSI	Open System Interconnection Model
PCS	Personal Communications Services
PDE	Possible Duplicate Emission (issued by Network)
PDIN	Public Domain Intelligent Network
PDM	Possible Duplicate Message (issued by BMS)
PKI	Public Key Infrastructure
POP	Point of Presence
PVN	Private Virtual Network
RS232C	Serial port and physical layer standard that defines 25 circuits.
RTGS	Real Time Gross Settlement
SCCP	Signaling Connection Control Part of SS7
SCE	Service Creation Environment
SCP	Service Control Point

SDN	Software Defined Network
SIPN	SWIFT Internet Protocol-based Network
SLA	Service Level Agreement
SMS	Service Management System
SNA	Systems Network Architecture
Software intelligence	Coded as programs, utilities, or modules and may reside in the active memories of computers during execution phase.
SPC	Stored Program Control. Capacity to control the network functions by programs or microcode generated and stored as software or utilities.
SS7	Signaling System 7
SSP	Service Switching Point
STN	SWIFT Transport Network
STP	Signal Transfer Point
STP	Straight Through Processing
SWIFT	Society for Worldwide International Financial Communications
SWIFTNet	SWIFT Next Generation IP-Network
TCAP	Transaction Capabilities Application Part of SS7
TCP/IP	Transmission Control Protocol/Internet Protocol
Telex	1 voice channel can contain 12 150-Hz telegraph channels or 24 50-bps Telex channels. International 50-bps service used over much of the world.
TFM	Transaction Flow Monitoring
TRN	Transaction Reference Number assigned by FI application systems.
TTS	Text To Speech synthesis
TWX	This service is a Western Union 150-bps teletypewriter service with dial-up connections like Telex.
VAN	Value Added Network
VPS	Virtual Private Network
Western Union	Operates a national telegraph message service to all parts of the United States. Western Union leases private communications links and operates 2 public dial-up telegraph networks, a Telex network compatible with the worldwide Telex network , and the Teletype Writer Exchange (TWX) network (which is bought from AT&T).
XML	Extended Markup Language

Trademarks

Symbol	Corporation
AIX	IBM
CICS	IBM
DB2	IBM
HP	Hewlett Packard
IBM	IBM
Java	Sun Microsystems
MERVA	IBM
MQSeries	IBM
OSI	Open Systems Interconnection
OS/390	IBM
RACF	IBM
Solaris	Sun Microsystems
SWIFT	SWIFT
Windows NT	Microsoft

Index

Advanced Intelligent Network, 1, 18, 19, 21, 22, 146, 279, 280, 281
File Transfer Protcol, 300
Financial Institution, iv, 1, 2, 3, 15, 17, 22, 23, 38, 144, 146, 147, 244, 256, 267, 279, 300
Intelligent Peripheral, iv, 1, 14, 17, 18, 21, 22, 23, 30, 31, 32, 146, 181, 256, 279, 280, 297, 301, 302
Markov chain, 44, 45
Poisson process, 46
Service Switching Point, 15, 21, 22, 144, 154, 234, 302
Service Transfer Point, 3, 15, 21, 154, 207, 280, 282, 302
Stored Program Control, 302
Transaction Reference Number, 80, 110, 113, 114, 115, 126, 127, 283, 284, 288, 289, 290, 291, 292, 293, 294, 295, 302
Transmission Control Protocol/Internet Protocol, 1, 181, 256, 280, 297, 302
X.25, 1, 22, 23, 31, 32, 241, 244, 259, 280, 297