Middleware for the SUNSAT Field Station

by

Andrew Robert Cardoza

A thesis submitted to the Department of Electrical and Electronic Engineering of the University of Stellenbosch in partial fulfilment of the requirements of Master of Science in Engineering..

Supervisor: Dr S. Mostert

December 2001

Declaration

I the undersigned hereby declare that the work in this thesis is my own original work and has not previously in its entirety or in part been submitted at any university for a degree.

Abstract

SRMA currently used on the SUNSAT field station is an effective MAC layer protocol but lacks several features. Sufficient support for flexible communication, acknowledgement of individual data packets, simultaneous communication of multiple instances from a higher layer over a single underlying layer, and a robust FTP procedure to transfer large amounts of data are some of the features that the SUNSAT field station should cater for.

ORBCOMM, GSM and WAP implementations are discussed. Some features from these implementations that are useful in the LEO communication environment are designed into an additional layer of software. This additional layer, called middleware, is proposed and uses SRMA services to deliver store-and-forward services. It supports high volumes of short transactions and session suspend and resume facilities between the SUNSAT satellite and field station.

Keywords: SUNSAT, Communication, Field station, WAP, GSM, ORBCOMM

Opsomming

Die SRMA pakketkommunikasie-protokol op SUNSAT se prototipe grondterminaal is voldoende om die oordrag van boodskappe oor die kommunikasieverbinding te reguleer. Dit is onvoldoende om fasiliteite soos kommunikasie van 'n aantal instansies in 'n hoër kommunikasie laag oor 'n enkele laer laag en om 'n robuuste data oordrag protokol vir groot hoeveelhede data van 'n grondterminaal te ondersteun.

ORBCOMM, GSM en WAP stelsel toepassings word bespreek. Resultate wat spruit uit die drie toepassings word gebruik om kommunikasie in 'n lae wentelbaan kommunikasie omgewing te bevorder. 'n Bykomende sagteware laag, bekend as middle-ware, word aanbeveel om voorafgenoemde funksionaliteit mee te implimenteer. Die middle-ware gebruik die beskikbare dienste van die SRMA pakketkommunikasie-protokol om toegang te verkry na die kommunikasie kanaal. Sodoende kan hoë volumes kort transaksies, en sessie-stop en hervat fasiliteite 'n werklikheid gemaak word.

Sleutelwoord: SUNSAT, Kommunikasie, grondterminaal, WAP, GSM, ORBCOMM

Table of Contents

I

Glossary

The following is a list of terms and abbreviations frequently used throughout the thesis. It serves as reference or lookup table.

ARQ	- **A**utomatic **R**epeat re**Q**uest
AX.25	- **A**mateur **X.25**
BBS	- **B**ulletin **B**oard **S**erver
BSC	-**B**ase **S**tation **C**ontroller
BTS	-**B**ase **T**ransceiver **S**tation
CSPDN	-**C**icuit-**S**witched **P**ublic **D**ata **N**etworks
DTE	-**D**ata **T**erminal **E**quipment
EIR	-**E**quipment **I**dentity **R**egister
ETS	- **E**uropean **T**elecommunications **S**tandard
FDMA	-**F**requency **D**ivision **M**ultiple **A**ccess
FTP	-**F**ile **T**ransfer **P**rotocol
GBN	- **G**o-**B**ack-**N**
GES	-**G**ateway **E**arth **S**tation
GMSC	-**G**ateway **M**obile **S**ervices switching **C**entre
GNCC	-**G**ateway **N**etwork **C**ontrol **C**entre
GSM	-**G**lobal **S**ystem for **M**obile communications
GSS	- **G**ateway Message **S**witching **S**ystem
HLR	-**H**ome **L**ocation **R**egister
HTTP	- **H**yper**T**ext **T**ransfer **P**rotocol

IMEI	- **I**nternational **M**obile **E**quipment **I**dentity
IMSI	- **I**nternational **M**obile **S**ubscriber **I**dentity number
IWF	- **I**nter**W**orking **F**unction
LEO	- **L**ow **E**arth **O**rbiting
ME	- **M**obile **E**quipment
MEO	- **M**edium **E**arth **O**rbit
MS	- **M**obile **S**tation
MSC	- **M**obile **S**ervices switching **C**entre
NCC	- **N**etwork **C**ontrol and management **C**entre
NMS	- **N**etwork **M**anagement **S**ystem
OCCNet	- **O**RBCOMM **C**ommand and **C**ontrol **Net**work
OMC	- **O**peration and **M**aintenance **C**entre
OMS	- **O**RBCOMM **M**essage **S**witch
ORBCOMM	- Two-way Satellite Communication System for Data-only messages.
PDN	- **P**rivate/Public **D**ata **N**etworks
PDU	- **P**rotocol **D**ata **U**nit
POTS	- **P**lain **O**ld **T**elephone **S**ervice
PSPDN	- **P**acket-**S**witched **P**ublic **D**ata **N**etworks
PSTN	- **P**ublic **S**witched **T**elephone **N**etwork
OQPSK	- **O**ffset **Q**uadrature **P**hase **S**hift **K**eying
RLP	- **R**adio **L**ink **P**rotocol
RPC	- **R**emote **P**rocedure **C**all
RTT	- **R**ound **T**rip **T**ime
RTXn	- **R**eal **T**ime e**X**ecutive **N**ano Kernel
SAP	- **S**ervice **A**ccess **P**oint
SC	- **S**ubscriber **C**ommunicator
SDL	- **S**ystem **D**escription **L**anguage
SIM	- **S**ubscriber **I**dentiy **M**odule
SMS	- **S**hort **M**essage **S**ervice
SMTP	- **S**imple **M**ail **T**ransfer **P**rotocol
SSID	- **S**econdary **S**tation **ID**entifier
SSL	- **S**ecure **S**ockets **L**ayer
SRP	- **S**elective **R**epeat **P**rotocol
SRMA	- **S**plit-channel **R**eservation **M**ultiple **A**ccess

SUNSAT - Stellenbosch **UN**iversity **SAT**telite

TDMA - **T**ime **D**ivision **M**ultiple **A**ccess

TA - **T**erminal **A**dapter

TCP - **T**ransmission **C**ontrol **P**rotocol

T/TCP - **T**ransmission **C**ontrol **P**rotocol with **T**ransactions

TE - **T**erminal **E**quipment

UHF - **U**ltra **H**igh **F**requency

UI - **U**nnumbered **I**nformation AX.25 Frame

URL - **U**niversal **R**esource **L**ocators

URI - **U**niversal **R**esources **I**dentifiers

VHF - **V**ery **H**igh **F**requency

VLR - **V**isitor **L**ocation **R**egister

WAE - **W**ireless **A**pplication **E**nvironment

WAN - **W**ide **A**rea **N**etwork

WAP - **W**ireless **A**pplication **P**rotocol

WCMP - **W**ireless **C**ontrol **M**essage **P**rotocol

WDP - **W**ireless **D**atagram **P**rotocol

WML - **W**ireless **M**arkup **L**anguage

WSP - **W**ireless **S**ession **P**rotocol

WTA - **W**ireless **T**elephony **A**pplication

WTLS - **W**ireless **T**ransport **L**ayer **S**ecurity

WTP - **W**ireless **T**ransaction **P**rotocol

WWW - **W**orld **W**ide **W**eb

II

List of Figures

Chapter 1

Introduction

Small, low-cost **LEO** (**L**ow **E**arth **O**rbit) satellites are increasingly integrated into the ever-expanding global communication network. LEO amateur satellites enhanced with on-board store-and-forward processing facilities, first developed during the 1980s, have laid the foundation and illustrated the concept's viability. The University of Stellenbosch has successfully developed, launched and commissioned South Africa's first operational satellite, named **SUNSAT** (**S**tellenbosch **UN**iversity **SAT**ellite).

SUNSAT is an educational satellite project aimed at increasing engineering research and design opportunities for graduate students. The other goals are to promote interest in technology through school participation programmes and to increase industrial and international interaction. This 64 kg micro-satellite was successfully launched in February 1999 from NASA's Vandenberg launching facility. SUNSAT consists of a number of sub-systems that perform various specialised functions such as telemetry, telecommand, imaging, communications, etc.

The SUNSAT field station is a small, compact, portable low-bandwidth transceiver capable of exchanging data with SUNSAT. The function of the SUNSAT field station is to transmit and receive data to and from SUNSAT. SUNSAT downloads the data at a later stage to the ground station or when requested to do so by the data recipient. Resources such as bandwidth, memory, processing cycles and power are far more limited on mobile devices compared to fixed devices. This holds true for the SUNSAT field station as well. A complete protocol stack must be designed to allow subscribers to efficiently transmit and receive data to and from SUNSAT taking bandwidth limitation and the short communication window into consideration. This protocol stack must be able to support a variety of different services such as datagram services, reliable data delivery services and file transfers. In order to meet these objectives, this thesis proposes the addition of several extra layers to the already existing layers on the mobile field station.

A protocol stack consists of different layers as shown in Figure 1-1. The OSI Reference Model is illustrated in Figure 1-1.A [45]. Next to the OSI Reference Model is an illustration of the existing communication protocol layers for SUNSAT field station. The RF equipment and the **SRMA** (**S**plit-channel **R**eservation **M**ultiple **A**ccess) protocol [18] have already been investigated and proposed as solutions for the physical and medium access layers respectively.

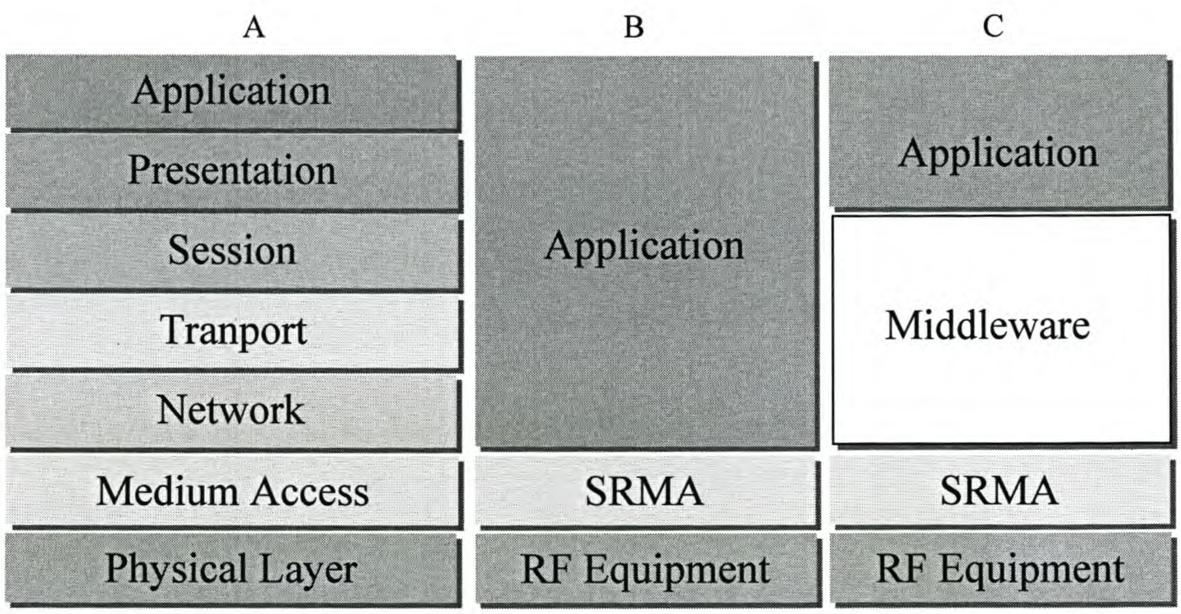

Figure 1-1 A :OSI Reference Model,B:Current field station,C:Proposed field station

In the current field station design, shown in Figure 1-1B, the application directly integrates with the SRMA layer. This thesis proposes the introduction of additional communication layers to enhance efficient resource utilisation while allowing for easier integration of applications with the field station.

Three different narrow-band wireless communicating systems have been designed, implemented and evaluated. The three different implementations were selected because of their similarities with the SUNSAT mobile field station. The similarities are:

- wireless communication to transfer data
- mobile and portable communication device is used
- functionality must be added to allow for effective communication over the network.

The three implementations could assist with the identification of key features required by mobile communication systems. From each implementation, a set of features promoting the stated objectives could be generated. Practical lessons learned here would be designed into a prototype middleware solution for the SUNSAT field station. Mobile middleware can be defined as the enabling layer of software that is used by application developers to connect their applications with different mobile networks and operating systems without introducing mobility awareness in the applications[49]. Optimization techniques, such as header compression, delayed acknowledgements and concatenation of several smaller packets into one may be incorporated into middleware to reduce the amount of traffic on the wireless networks.

Chapter 2 discusses the DataTREK communication system that was developed for use on the ORBCOMM Network. ORBCOMM is the world's first commercial, global, two-way data-only message communication system. The **SC** (**S**ubscriber **C**ommunicator) is ORBCOMM's equivalent to the SUNSAT field station.

GSM (**G**lobal **S**ystem for **M**obile communications) is one of the leading mobile, terrestrial-based networks. Fleet management and vehicle tracking systems are increasingly using GSM services to transport critical information between the vehicle and the central management station. A communication protocol stack was designed to introduce additional functionality to effectively use GSM services while supporting fleet management functionality. The design of the DX Protocol stack is discussed in detail in Chapter 3.

WAP (**W**ireless **A**pplication **P**rotocol) specifies the framework and the network protocols for wireless devices such as mobile telephones, pagers and personal digital assistants. Its main objective is to create a global wireless protocol specification that will work across different wireless network technologies [50]. WAP features allow mobile terminals to efficiently use resources on narrow-band, high-latency networks. The SUNSAT store-and-forward facility is a narrow-band, high-latency service as well and may therefore benefit from techniques used in WAP for effective resource utilisation.

The proposed solution, including its drawbacks and advantages, are then discussed.

Chapter 2

Data Collection System

The South African Agricultural Research Council has a network of monitoring stations distributed across South Africa. These stations assist with weather and crop disaster forecasts. Collected data is transferred via **PSTN** (**P**ublic **S**witched **T**elephone **N**etwork) and GSM communication networks from remote monitoring stations. This data collection procedure is limited to regions with PSTN or GSM infrastructures. Data collection from monitoring stations situated in areas with a lack of communication infrastructure is expensive, labour intensive and time consuming. A more cost-effective solution is needed that allows for easy data collection from stations situated in remote, difficult-to-reach areas where there is a lack of communication infrastructure.

Store-and-forward LEO satellites have shown that data can be transported between virtually any two locations on earth. Since then several commercial satellite communication operators of which some are still in their concept stages, have come to the forefront. Iridium, ICO, Teledesic, GlobalStar, Odyssey, Skybridge, ORBCOMM and Celestri use or plan to use orbiting satellites in their satellite constellation segment. The ORBCOMM communication network has shown itself to be an effective transport medium for small amounts of data. The OBRCOMM network can thus be used to transport data from the remote monitoring stations situated in areas lacking in terrestrial-based network infrastructures.

A brief introduction to some of these satellite networks is given in the next section. This is followed by a detailed discussion of the ORBCOMM System. An operational description of the DataTREK Global Data Collection System follows. The DataTREK System is a typical example of what a store-and-forward application entails and could therefore assist with the identification of features used in such systems.

2.1 Commercial Satellite Systems

- Motorola's **Iridium** comprises 66 satellites at a nominal altitude of 780 km in six polar orbital planes. This satellite constellation is supported by a network architecture similar to cellular systems. Satellites form a network using 23 GHz cross-links that are linked to a system control facility, gateways and subscriber units [25]. Direct, seamless point-to-point subscriber access is possible from anywhere in the world independently of the PSTN.

- **Teledesic,** with a constellation of 288 LEO satellites with optical inter-satellite links, is the most ambitious of the proposed broadband systems [43]. Teledesic terminals can provide bandwidth on demand of up to 64 Mbps from virtually anywhere on the globe at service levels comparable to terrestrial fibre. The large number of satellites results in a high elevation angle, which will reduce the effect of interference from high obstacles and rain attenuation. Its aim is to provide high-speed access to the Internet by 2003.

- The **Globalstar** System is based on a constellation of 48 LEO satellites, orbiting in eight planes at 52 degree inclinations, at an altitude of 1406 km, with an orbital period of 113 minutes [25]. Calls are routed between a user terminal, via satellite, to a network of gateway ground stations. Any one of these ground stations is capable of serving users within a radius of 1000 km. GSM/Globalstar dual-mode communicators allow for interaction with GSM networks when inside GSM coverage areas and otherwise via the Globalstar system.

- **Odyssey** consists of a constellation of 12 satellites orbiting in three planes (four satellites per orbit inclined at 55 degrees) at a higher medium earth orbit of 10354 km [25]. The Odyssey constellation provides continuous global coverage with dual satellite visibility in some major regions. Voice and data services are delivered to Odyssey mobile subscriber communicators via the Odyssey ground segment that is connected to the PSTN through the Odyssey Earth Stations and local gateways.

- **Skybridge** will ultimately be an 80 LEO satellite constellation with broadband capacity to provide high-speed local access, multimedia and interactive services. Satellites are used in the simple bent-pipe configuration which limits switching and routing to the gateway. Space links therefore have lower operating costs and lower risks [43].

2.2 ORBCOMM System Architecture

ORBCOMM is the world's first commercial, global, two-way data and message communication system. It is a data-only system using a digital packet protocol. With data rates of 4,8 Kbps its primary markets are asset tracking, personal messaging, data collection and remote control of semi-autonomous to autonomous systems. It comprises a constellation of LEO MicroStar satellites, ORBCOMM Gateways, **GNCC** (**G**ateway **N**etwork **C**ontrol **C**entre), **NCC** (**N**etwork **C**ontrol and management **C**entre) and small portable VHF transceivers called SCs. These network components, displayed in Figure 2-1, are briefly discussed in the following subsections.

2.2.1 Subscriber Communicators

An SC is a wireless VHF transceiver that transmits subscriber messages to the ORBCOMM System. Messages may be delivered to another SC or devices connected to the Internet. These SCs are ORBCOMM's equivalent to the SUNSAT mobile station. SCs adhere to the *ORBCOMM Air Interface Specification*, *Subscriber Communicator Specification* and *ORBCOMM Serial Interface Specification* when a RS232 port is available [31].

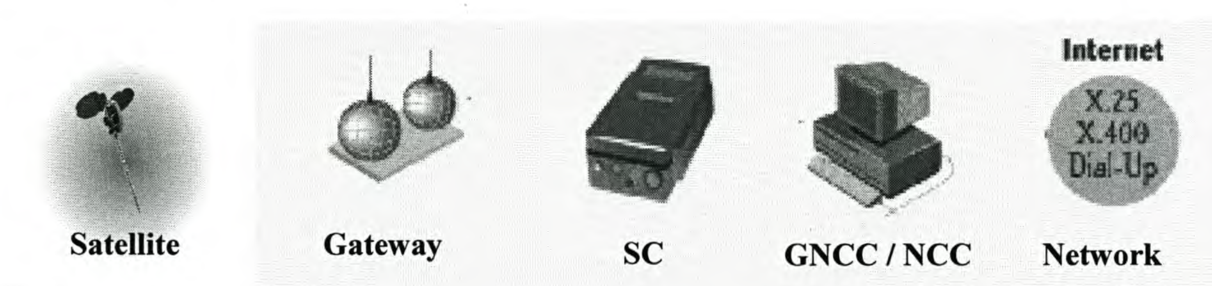

Figure 2-1 ORBCOMM System Components

2.2.2 MicroStar Satellites

The ORBCOMM System would ultimately consist of up to 36 Orbital MicroStar satellites launched into six orbital planes. The main constellation consists of four orbital planes (planes A, B, C and D) of eight satellites each [32]. The first three planes are inclined at 45° with respect to the equator (satellites at a target altitude of 825 km) and are spaced 45° apart. The fourth plane of eight satellites (plane D), is planned to be inclined at 0° (equatorial orbit) at an 825 km altitude and spaced 45° apart. Two supplemental orbital planes [planes F and G], containing two satellites each, provide coverage from approximately 780 km altitude and are spaced 180° apart. Plane F is inclined at 70° and plane G is inclined at 108°.

The communication subsystem, which is the principal payload, consists of four major parts, namely [32]:

- **The subscriber communication section** consists of one subscriber transmitter, seven identical receivers, the associated *receive and transmit* filters and antennas. Normally, six of the receivers will be used as subscriber receivers and the seventh will be used as a receiver to identify noise-free channels. Data is uplinked and downlinked at data rates of 2400 bps and 4800 bps respectively.

- The **ORBCOMM Gateway Communication section** consists of a dedicated transmitter and receiver contained in a single package. The 57.6 kbps downlink signal to the GES is transmitted using an **OQPSK** (**O**ffset **Q**uadrature **P**hase **S**hift **K**eying) modulation in a **TDMA** (**T**ime **D**ivision **M**ultiple **A**ccess) format.

- The **Satellite network computer** receives the uplinked packets from the subscriber and the ORBCOMM Gateway receivers, distributes them to the appropriate transmitter, identifies clear uplink channels and interfaces with the GPS receiver to extract information pertinent to the communications system.

- The **UHF transmitter** is a specially constructed 1 Watt transmitter that is designed to emit a highly stable signal at 400.1 MHz.

2.2.3 Gateways

The principal role of the ORBCOMM Gateway is to provide message processing and subscriber management for a defined service area [32]. It acts as interface between the subscriber and the interconnected **PDNs** (**P**rivate/**P**ublic **D**ata **N**etwork) and PSTN. Each ORBCOMM Gateway incorporates four major subsystems:

- The **GSS** (**G**ateway Message **S**witching **S**ystem) performs the actual message processing of the proprietary ORBCOMM protocol, does routing and conversion to and from X.400 and **SMTP** (**S**imple **M**ail **T**ransfer **P**rotocol).

- The **GES** (**G**ateway **E**arth **S**tation) provides a RF communication link between the GSS and the Satellite constellation.

- The **NMS** (**N**etwork **M**anagement **S**ystem) provides health and status management functions, monitoring of message throughput and monitoring of GES-GSS inter-connecting links.

- **OCCNet** (**O**RBCOMM **C**ommand and **C**ontrol **Net**work) is the world-wide **WAN** (**W**ide **A**rea **N**etwork) network that connects all ORBCOMM Gateways.

2.2.4 Network Control Centre

The NCC manages the complete ORBCOMM System through the world-wide interconnected network OCCNet. It monitors message traffic for the entire ORBCOMM System, manages all message traffic that passes through the US ORBCOMM Gateway and processes and analyses all satellite telemetry. This facility, located at Dulles (Virginia, USA), is staffed 24 hours a day, 365 days a year.

2.2.5 Service Features

There are four basic service elements that the ORBCOMM System is capable of providing, namely [32]:

1. **Data Report**: This is the basic service element whereby an SC generates a short report (a single packet containing less than or equal to six (6) bytes of user defined data) that is transmitted via the random access protocol.

2. **Message:** This is the basic service element whereby a longer sequence of data is transferred to or from an SC. Message lengths are typically less than 100 bytes, although the ORBCOMM System can handle messages that are longer in length.

3. **GlobalGram:** This is a single, self-contained data packet that an SC sends or receives to/from a satellite when that satellite does not have access to an ORBCOMM Gateway. A maximum number of 229 user data bytes may be carried within a GlobalGram^{TM}.

4. **Command:** This is the basic service element whereby a short command (single packet containing less than or equal to five (5) bytes of user-defined data) is transmitted to an SC.

2.2.6 ORBCOMM Data Flow

Figure 2-2 shows the bi-directional flow of user data through the ORBCOMM System. A user sends data to the SC. The SC stores the data until a satellite appears. Data transmitted to the satellite is stored if there is no visible GES. Data is stored until the destination GES communicates with the satellite. This data is then downloaded to the gateway and forwarded via private and public networks (X.25, Internet) to its destination.

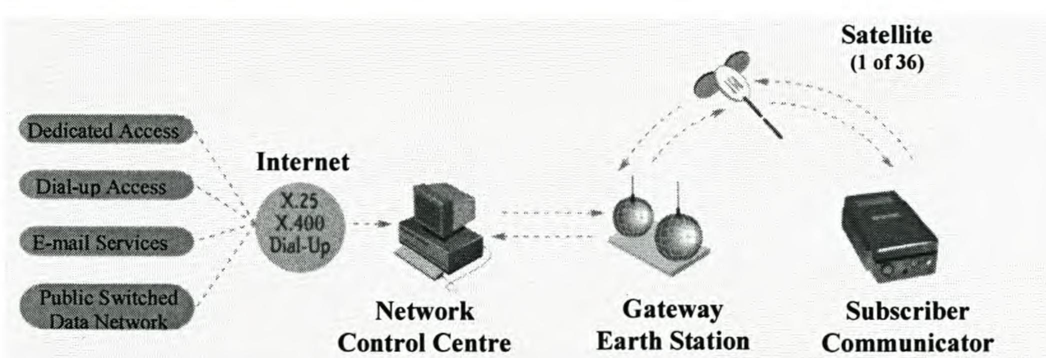

Figure 2-2 ORBCOMM Network Architecture

When a subscriber wants to forward data to a device connected to an SC, data is sent to the (G)NCC from whence the GES will send it to the selected or first available satellite. The satellite stores the data until a connection is established with the destination SC. Data is directly relayed to the SC when the SC and the GES share the same satellite footprint.

2.3 ORBCOMM Protocol

The ORBCOMM Communication Protocol Suite consists of the *ORBCOMM Air Interface Specification* and the *ORBCOMM Serial Interface Specification.* ORBCOMM subscriber units must conform to these specifications.

The *ORBCOMM Air Interface Specification* describes the communication protocol between the SC and ORBCOMM MicroStar LEO Satellites. The *ORBCOMM Serial Interface Specification* describes the communication protocol between the SC and the **DTE** (**D**ata **T**erminal **E**quipment) as illustrated in Figure 2-3.

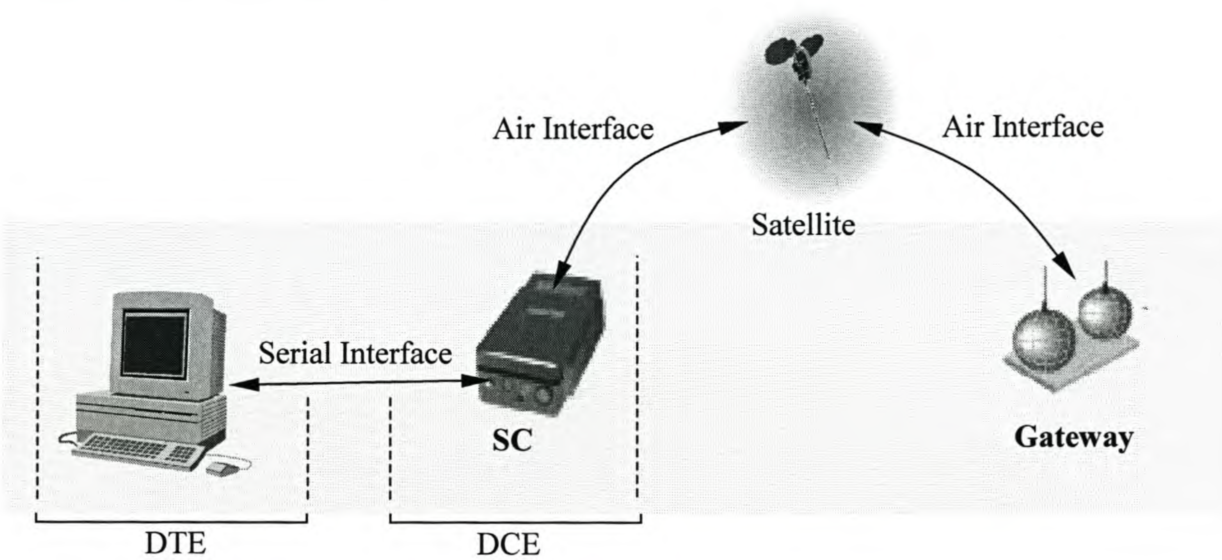

Figure 2-3 ORBCOMM Interfaces

Once power is applied to the SC, its RF receiver automatically searches for a satellite signal using an internally stored list of satellite downlink channels. If the SC has not detected a satellite signal at any of its stored channels, it continues to search for a satellite signal in all possible channels in the 137 to 138 MHz satellite-to-subscriber band. When the SC still does not find a downlink signal, it reverts to a reduced power (sleep) mode for a predefined time interval, then re-attempts the search process. The satellite signal contains the control information necessary to initiate a data transfer session. This control information includes timing information, the connected ORBCOMM Gateways and the current available uplink random access channels. The satellite re-transmits this information at least once approximately every 8 seconds.

Once the SC receives the control information signal from the satellite, it initiates the **A/C** (**A**cquire/**C**ommunicate) procedure by sending a *request to send* message to the satellite. The satellite should receive the burst correctly in the absence of interference or time-overlapping bursts on the same receive channel. It responds with control information specific to that SC. The SC will automatically re-transmit the request if it does not receive an assignment or an acknowledgement within a specific period.

The A/C process is executed when SC-Originated GlobalGrams are queued on the SC waiting for transmission to the first available satellite. The satellite responds with an assignment message whereby the SC sends the GlobalGram on the assigned frequency. Sufficient time is reserved for the transmission of the number of bytes indicated in the request packet. The satellite then sends an inbound GlobalGram acknowledgement packet. The satellite stores the GlobalGram until the destination ORBCOMM Gateway connects to the satellite.

SC requests to transfer larger amounts of data are acknowledged with assignment messages, containing the number of packets and their length, from the ORBCOMM Gateway. The satellite reserves a time slot and an uplink frequency channel. The assignment is complete when this information is transmitted to the SC. The SC transmits the message burst at the assigned time and assigned frequency and then waits for an acknowledgement from the gateway. This process is repeated until the message is successfully transferred from the SC to the GSS.

The protocol is designed for automatic recovery from events such as incorrectly received packets or satellites going out of view. Message packets may be transferred over different ORBCOMM Satellites. The ORBCOMM Gateway puts the message together and transfers the complete message.

2.4 DataTREK Data Collection System

Data loggers are digital devices that receive analogue data, do analogue-to-digital data conversion, data processing, storage of the data and ultimately the transmission of this data via available serial communication ports. Communication devices such as PSTN modems, GSM modems, computers or any other communication devices may be connected to the serial port.

The ORBCOMM System was selected as transport medium to combat the lack of communication infrastructure in those remote regions where monitoring stations are located. The ORBCOMM System is perfectly suited to periodically transport small amounts of data. The requirements for the DataTREK Data Collection System are:

- The data collection procedure must be comfortably executed from a desktop computer.
- These data loggers are not capable of directly communicating with an ORBCOMM SC. An interfacing device must be designed that requests data from the data logger and transmits data in the form of GlobalGrams to the SC. It is called the DataTREK protocol gateway.
- Since the DataTREK protocol gateway would operate in remote areas for long uninterrupted periods of time it must be designed to be robust and to operate in extreme conditions.
- Data compression should be incorporated to minimize data transmission costs.

2.4.1 System Architecture

The system architecture, illustrated in Figure 2-4, shows all the relevant system components. This figure shows how the Protocol Gateway completes the communication path between the end-user and a data logger.

Since no ORBCOMM GES is present to service the Southern Africa region, the system was designed to operate using only ORBCOMM's GlobalGram services. The DataTrek protocol gateway periodically retrieves data from the logger, does data compression and forwards the compressed data to the SC. The SC transmits the data to the first available ORBCOMM Satellite. Data is downlinked to a selected ORBCOMM Gateway, forwarded to the NCC from whence it is forwarded to a E-mail server [27]. This E-mail server forwards the data to the respective users. This complete END-to-END solution uses locally designed end-user software running on the Windows 95/ NT operating system. It processes collected data and stores it on a local database.

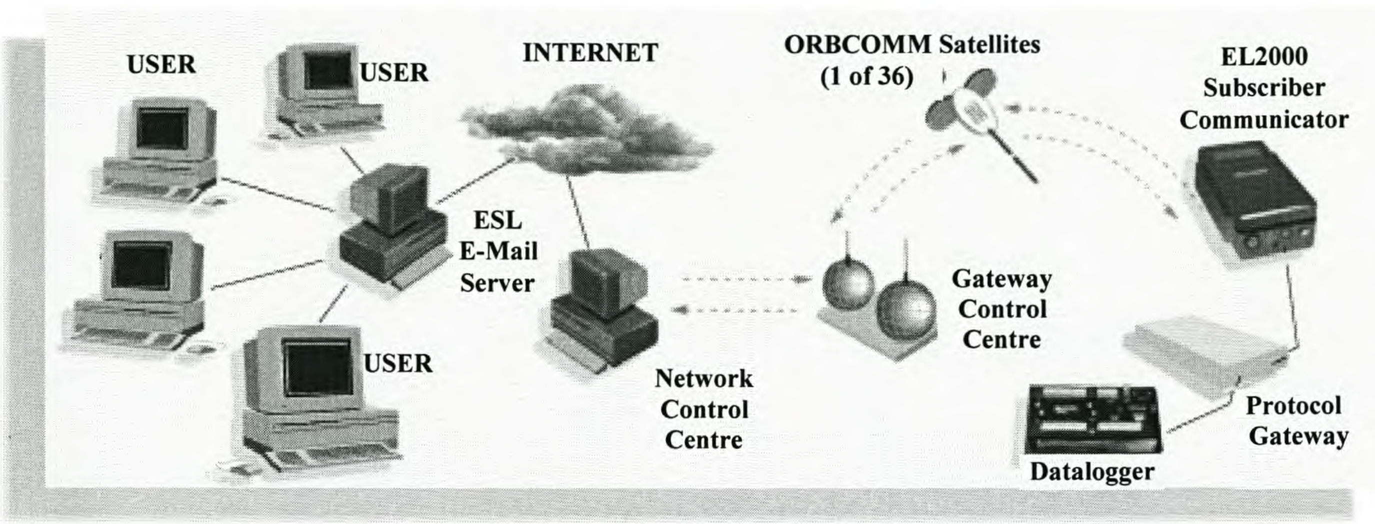

Figure 2-4 DataTREK System Architecture

The SDL Diagram in Figure 2-5 is equivalent to Figure 2-4. See [13] for an introduction to SDL. It shows the data flow between the different components of the DataTREK System. The protocol gateway sends a command to the logger using a data packet format recognisable by the data logger. The data logger's system time, request for logger information, request for data and deletion of data are all issued using packet formats recognisable by the specific data logger. Data transmitted from the protocol gateway to the SC is in the form of GlobalGrams. The SC acknowledges a correctly-received GlobalGram. Data between the SC and protocol gateway conforms to the ORBCOMM's Serial Specification [31]. The SC forwards these GlobalGrams to passing satellites. Data between the SC and ORBCOMM satellite conforms to the Air Interface Specification [30]. The ORBCOMM System delivers data in the form of E-mail messages. The data is contained in the main body of the E-mail message. These E-mail messages are delivered to the DataTREK E-mail server. Data is then forwarded to the appropriate subscribers.

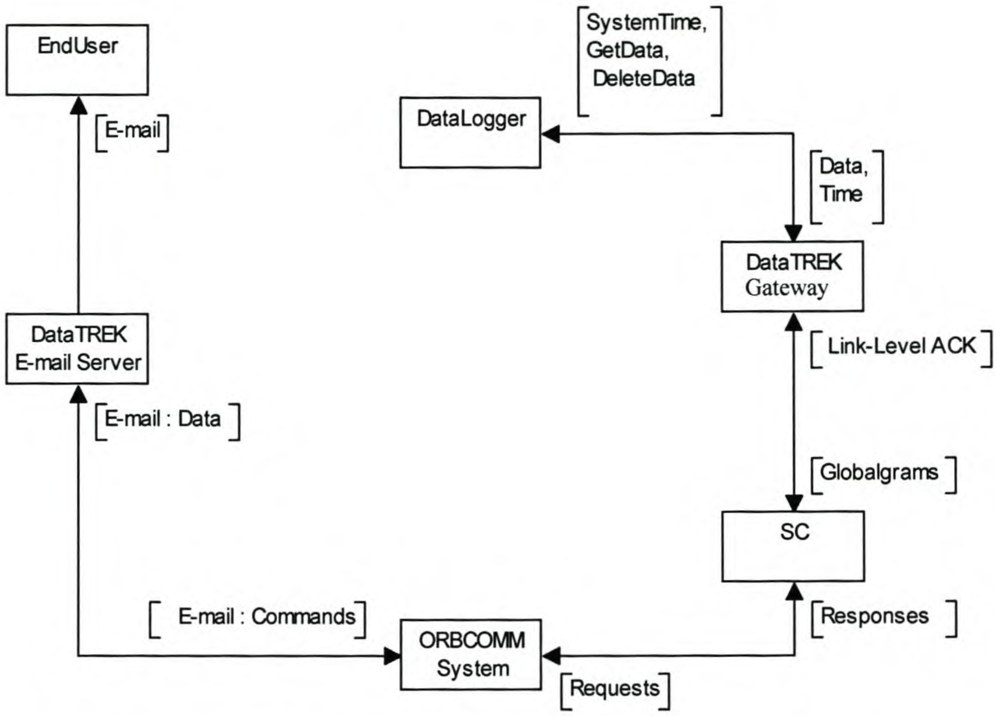

Figure 2-5 SDL System Level Data Flow of DataTREK System

2.4.2 DataTREK Protocol Gateway

The functional block diagram of the DataTREK Protocol Gateway is illustrated in Figure 2-6. Over-voltage and over-current equipment protect against lightning and short circuits.

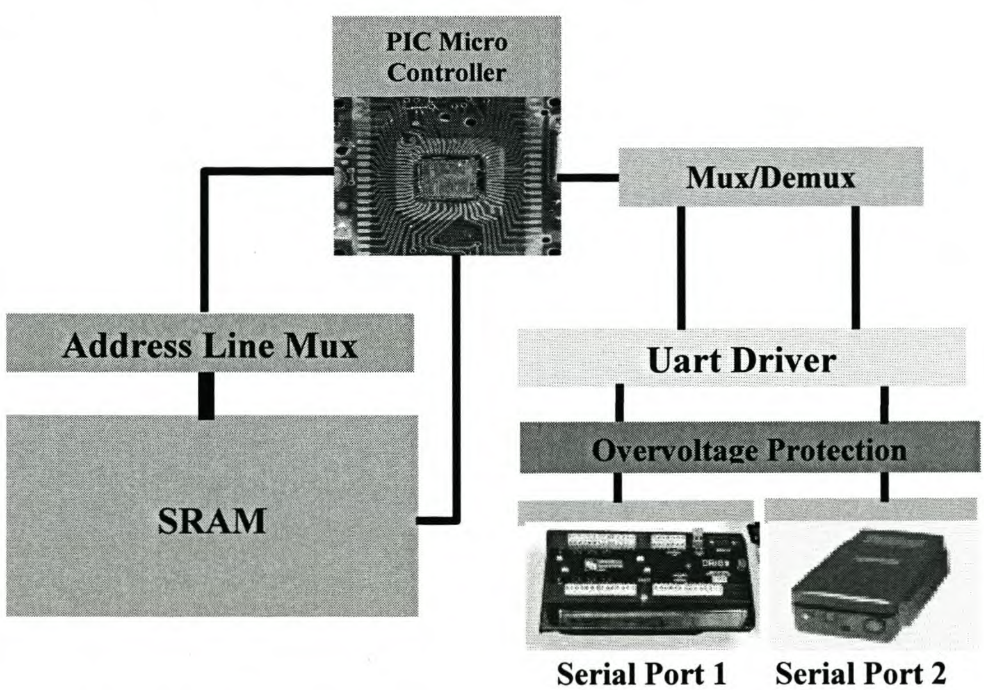

Figure 2-6 Block Diagram of DataTREK Protocol Gateway

At the heart of the DataTREK Protocol Gateway is a PIC16C74A micro controller that is responsible for controlling peripheral devices such as the SRAM. The DataTREK Protocol Gateway has two serial ports that are emulated in software onto one physical UART communication port. The protocol gateway initiates communication with the data logger by periodically requesting data from the logger.

Programmed data compression is used since the most popular commercially used Lempel-Ziv algorithms are only effective for files bigger than 1 Kbyte. GlobalGrams are limited to 229 user bytes. A compression ratio of less than 1/3 (compressed file = 1/3 of original file) delivered by programmed compression is far more competitive than Lempel-Ziv algorithms and their variants, which form the heart of most compression procedures. The result is 8-bit binary data that must be delivered to the end-user via E-mail messages. This data is encapsulated within the main body of E-mail messages. Only 7-bit data can be stored in the E-mail main body. An encoding procedure, transforming 8-bit to 7-bit bytes, must therefore be performed. This encoding procedure is reversed with a similar decoding procedure at the end-user before data is stored. The SDL Block Diagram in Figure 2-7 is equivalent to the block diagram illustrated in Figure 2-6.

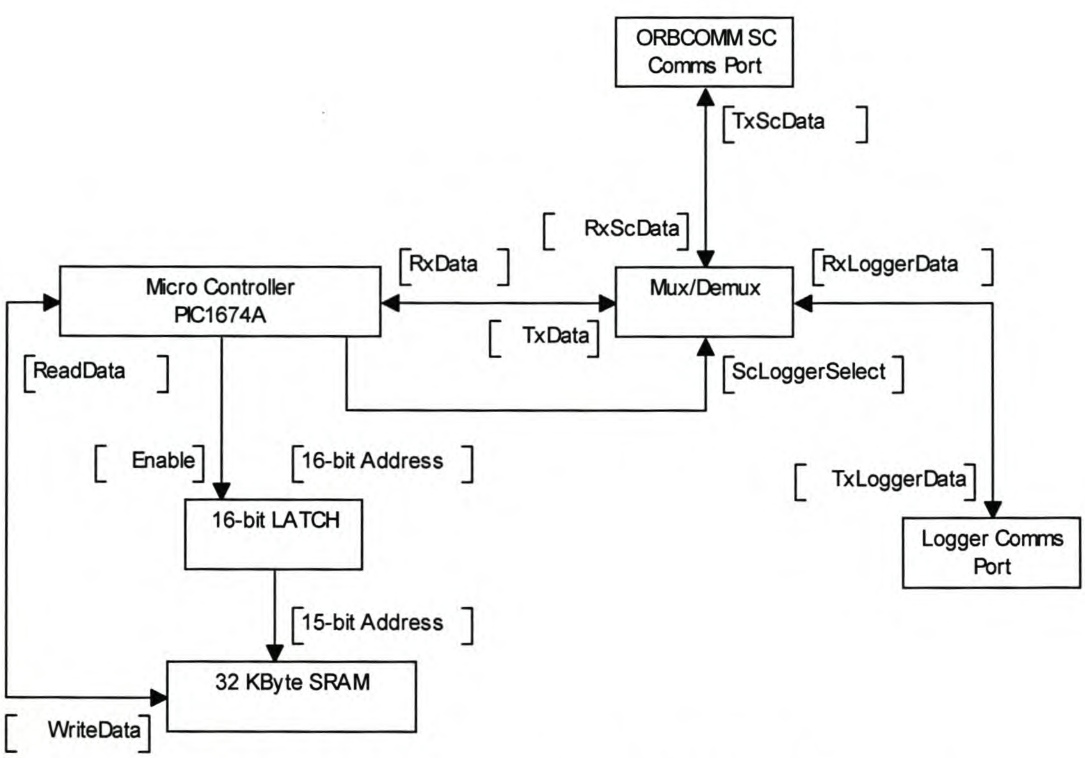

Figure 2-7 SDL Block Diagram of DataTREK Protocol Gateway

The micro-controller is programmed to use one 8-bit data port and a 16-bit latch (two octal latches) to address 32 Kbytes of SRAM. Data received from the data logger is compressed and temporarily stored in the SRAM. This data is read from the SRAM when it is forwarded to the SC. A multiplexer and demultiplexer are used in conjunction with software to emulate the two serial ports on the micro-controller's single UART port.

2.5 System Recommendations

The DataTREK system has been implemented in several remote weather monitoring stations. The OSI protocol stack and the protocol gateway are illustrated in Figure 2-8. On the left the functionality of the DataTREK Protocol Gateway and the ORBCOMM SC are mapped against the OSI Protocol stack.

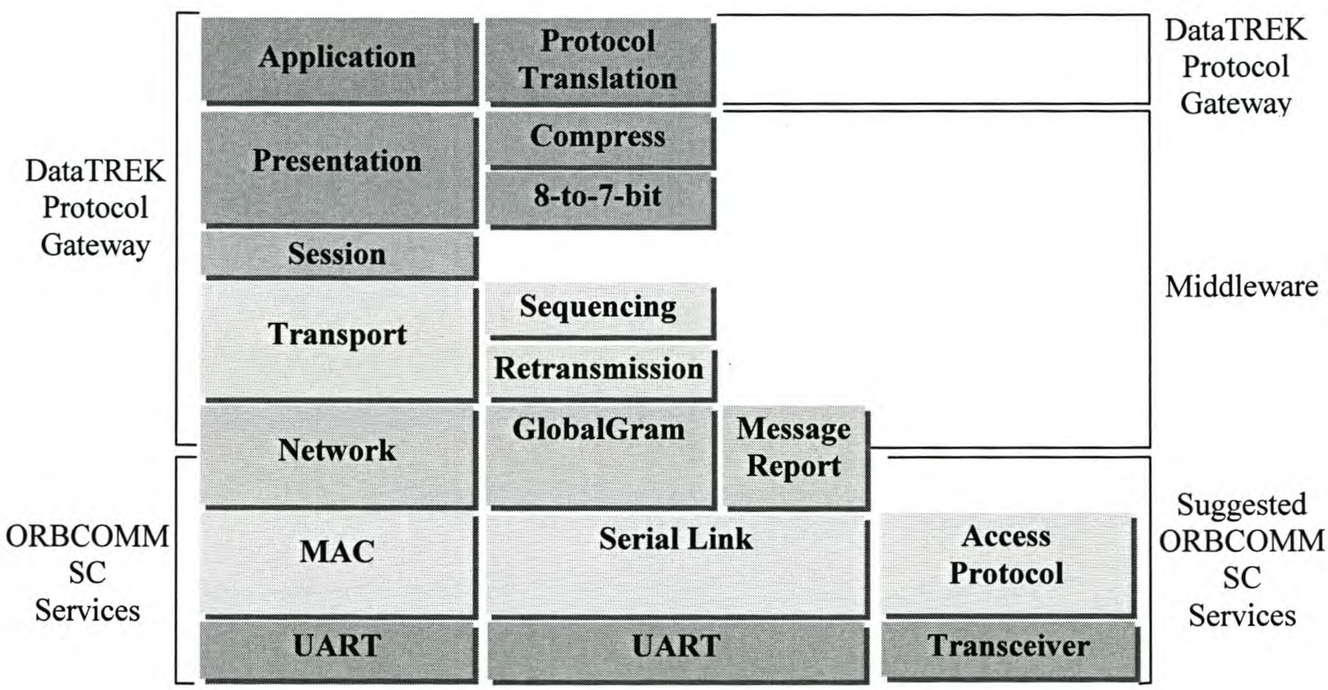

Figure 2-8 Protocol Stack comparison between OSI Stack and Protocol Gateway

The following remarks and recommendations can be made regarding the current implementations:

- ORBCOMM's GlobalGram facility is a true datagram facility and it is therefore representative of network layer functionality as illustrated in Figure 2-8. The GlobalGram service does not guarantee delivery e.g. loss of GlobalGrams due to system malfunction.

- ORBCOMM's GlobalGram service does not guarantee in-sequence data delivery. Out-of-sequence delivery may occur when an SC sends GlobalGrams to two different satellites. The second satellite may pass over the destination gateway before the first satellite and therefore has the opportunity to forward its data before that of the first satellite.

- Data is delivered in the main body of E-mail messages. This restricts content in the main body to 7-bit data. An 8-to-7-bit conversion facility added to the translator allows the transportation of binary data via e-mail messages. A 7-to-8-bit conversion facility added to the end-user's system management software allows conversion of e-mail message content to its original binary form again.

- The destination ORBCOMM gateway must be specified within the GlobalGram. This assumes that the subscriber is knowledgeable on the location of the closest GES. An SC that roams internationally may not always be optimized to use the closest GES.

ORBCOMM's Air Access MAC Protocol is similar in nature to SUNSAT's SRMA. These similarities and differences will be highlighted in chapter 6. Performance and data throughput in the ORBCOMM System and SUNSAT will therefore compare well with each other if differences are ignored.

The four different data messages (data reports, messages, GlobalGrams, commands) allow variety so that the most suitable message type for a specific implementation and environment may be selected.

Table 1 ORBCOMM Services Overview

ORBCOMM Service	Latency	Rate	Destination	Applications
GlobalGram	Minutes to Hours	Maximum of 229 user octets every 2-3 seconds.	Determined at transmission time	Very low data rate applications

Messages	Tens of Seconds	1200 bps	Determined at transmission time	Low data rate streaming applications.

ORBCOMM SCs already have additional functionality that allows applications to operate more independently from network services; like

- storage of data on the SC until a satellite passes; or
- storage of data received from the satellite until the application retrieves it.

Several key services, essential to the success of the DataTREK system, but lacking on the SCs are:

- sequencing information to the ORBCOMM messages to distinguish between different messages; retransmissions and sequencing.
- recovery of lost ORBCOMM messages by retransmission
- encoding/decoding of binary data transported in the main body E-mail messages; and
- introduction of loss-less compression algorithms specifically geared for small data portions .

Two octets are needed to represent these additional services sufficiently e.g.

1. R = 1 if GlobalGram is retransmitted and R=0 otherwise
2. 7-Bit sequence number that is incremented for each newly issued GlobalGram
3. E = 1 if Encoding is used and E=0 otherwise
4. 3-bit value relating to encoding type (0=8-to-7 bit translation, etc.)
5. C=1 if compression is used and C=0 otherwise
6. 3-bit value relating to compression type

Sequence Control		Content Encoding/Decoding			
R	7-bit sequence number	E	3-bit Encoding Type	C	3-bit Compression Type

The addition of 2 extra octets of administration information to a GlobalGram does not influence the efficiency dramatically. If $\eta_{GlobalGram}$ is taken as the original efficiency of a GlobalGram then the efficiency of the new GlobalGram with 2 octets of additional administration data, then

$$\eta_{NewGlobalGram} = \frac{data}{data+2}\eta_{GlobalGram} \text{ where data} \leq 227$$

This does not have a significant decrease in efficiency for larger GlobalGrams. The additional information thus results only in a small decrease in efficiency compared to the additional functionality and control gained.

2.6 Conclusion

The DataTREK System has been implemented successfully using the ORBCOMM GlobalGram services. The DataTREK Protocol Gateway connects the data logger to the SC. The SC Serial Interface, data compression and 8-to-7-bit translation is implemented in the DataTREK Protocol Gateway. The Desktop and end-user application receives data via the **POP3** (**P**ost **O**ffice **P**rotocol **3**). Before data is stored onto the private database, a 7-to-8-bit translation and decompression is done.

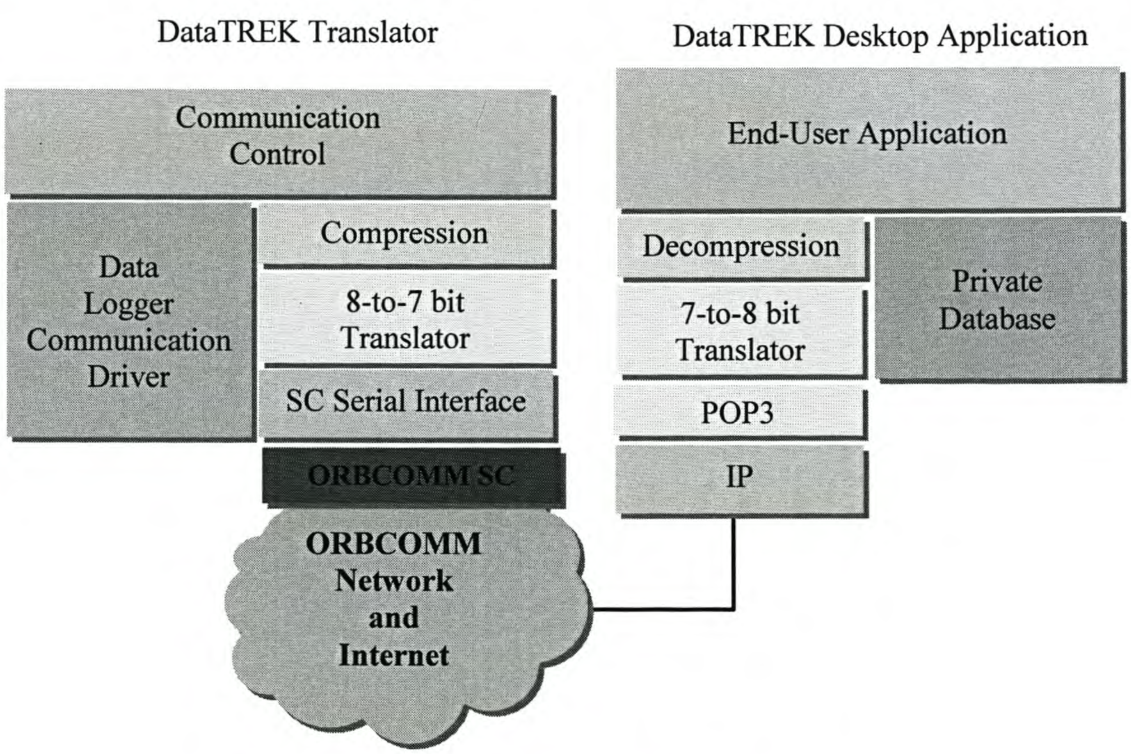

Figure 2-9 End-to-End DataTrek System Diagram

Since GlobalGrams are ORBCOMM's equivalent of datagrams, additional information is needed to do retransmissions, sequencing, data compression and data conversions. This functionality was originally added to the DataTREK implementation. One solution would be to add an additional layer of software that provide these services to the upper layer services such as the Protocol Gateway. A small number of control bytes and a small loss in efficiency must be sacrificed to gain benefits such as sequencing and acknowledgements.

ORBCOMM is a prime example of how store-and-forward services may be provided to subscribers world-wide irrespective of their location. The SUNSAT field station would operate in a similar fashion as the ORBCOMM SC. DataTREK is a good example of a typical system that may benefit from store-and-forward services provided by LEO constellations. The successful implementation of the compression and 8-to-7-bit converter have directly contributed to the success of this project. Compression and the 8-to-7-bit converter, although small, made the project into an economic viable one. This should form part of an additional layer of software that connects the top-layer to the lower communication layers.

Chapter 3

GSM fleet management

GSM (**G**lobal **S**ystem for **M**obile communication) is a very popular second-generation mobile communication network. Compact and relatively inexpensive GSM handsets are now in everyday use and directly contribute to the success of GSM. Fleet management systems increasingly harvest GSM technology for remote vehicle monitoring. The fleet management system consists of two functional units namely; a fleet management base station and numerous mobile stations. GSM services such as **SMS** (**S**hort **M**essage **S**ervice) and circuit-switch data connections transport data between these functional units. Additional functionality is needed to turn a basic GSM modem into an intelligent fleet management device that uses the relevant GSM services efficiently and intelligently.

SUNSAT's store-and-forward services and the GSM fleet management system are quite similar. In both instances a mobile communicating unit (e.g field station versus mobile station) needs to transport data to another unit (e.g. ground/field station versus base station) using the available infrastructure (e.g SUNSAT/SRMA versus GSM). Since the GSM fleet management system and SUNSAT/SRMA store-and-forward services are similar in nature, features of the fleet management mobile station could be incorporated into the SUNSAT field station design.

A prototype protocol stack capable of supporting a range of fleet management functions is proposed and implemented. This chapter describes the development and implementation of such a scalable communication protocol. The system requirements are briefly described in the following section. This is followed by an overview of GSM, its services and architecture. A complete description of the prototype system is then given followed by the conclusion.

3.1 System Requirements

A scalable GSM communication protocol stack must be developed for implementation on the main fleet management system components. The fleet management system has the following system components as illustrated in Figure 3-1. The prototype protocol solution must be demonstrated on the following two units;

- **A mobile unit**: It is a GSM, wireless and autonomous embedded communicating device that is installed in a vehicle. The protocol stack should support reporting of system status, errors and the bi-directional transfer of large portions of data.

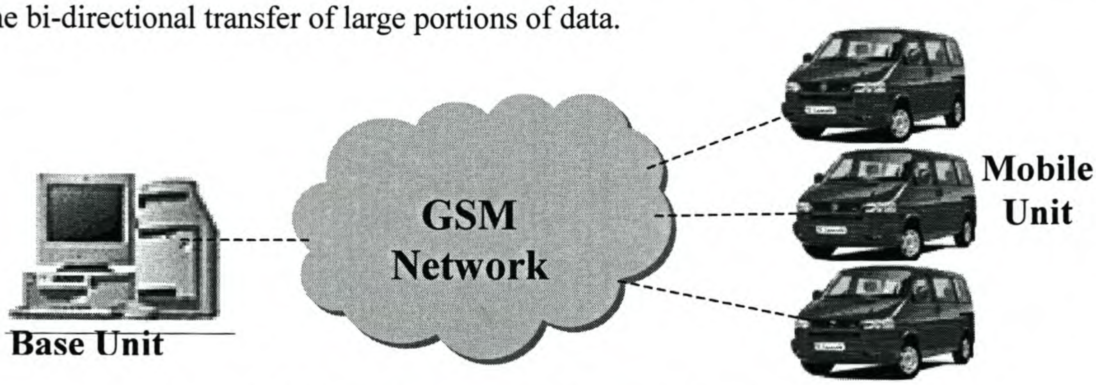

Figure 3-1 Fleet Management System Compone...

- **A Base unit**: It performs management operations, records and responds to messages from mobile units.

The prototype protocol stack should meet the following requirements:

- It should be scalable to allow for easy extension, architectural modifications and changes in functionality. It must be demonstrated on the mobile and base units without any modification.

- The mobile unit must be able to receive and transmit messages to report configuration settings, current status, errors, violations, etc.

- The base unit should be able to transmit messages to any of the mobile units.

- The mobile unit must be able to support GSM circuit-switched data connections with the base unit to allow exchange of large quantities of data.

- A robust **FTP** (**F**ile **T**ransfer **P**rotocol) must be designed to operate in conditions such as poor connections, a congested network and moving mobile units. Bandwidth resources must be efficiently utilised and an explicit restart mechanism must be used when data connections are lost.

- The mobile unit should be able to receive and transmit messages during data transfer.

- The top managing layer should seamlessly communicate with the base unit over both a GSM wireless link and a physical RS232 link.

3.2 GSM Services

GSM communication services can be divided into bearer services, teleservices and supplementary services. Telephony is the most basic teleservice supported by GSM. As with all other communications, speech is digitally encoded and transmitted through the GSM network as a data stream. A variety of data services are offered. GSM users can send and receive data, at rates up to 9600 bps, to/from **POTS** (**P**lain **O**ld **T**elephone **S**ervice), ISDN, Packet Switched Public Data Networks and Circuit Switched Public Data Networks users.

A unique feature of GSM is SMS. SMS is a bi-directional service allowing transport of short alphanumeric messages. Messages are transported in a store-and-forward fashion. A point-to-point SMS can be sent to another subscriber. SMS can also be used in a cell-broadcast mode, for sending messages such as traffic or news updates.

Supplementary services are provided on top of teleservices or bearer services. Phase I GSM networks provide several forms of call forwarding, and call barring of outgoing or incoming calls. Phase II GSM network services include caller identification, call waiting and multiparty conversations.

3.3 GSM Network Architecture

GSM is an evolving digital cellular radio standard that combines essentially the **TDMA** (**T**ime **D**ivision **M**ultiple **A**ccess) and **FDMA** (**F**requency **D**ivision **M**ultiple **A**ccess). It features international roaming, high security levels, superior speech quality and low terminal and subscriber service costs. The different entities in the GSM system are as follows [40]:

- The **MS** (**M**obile **S**tation) is the actual communication device and is the best known part of the cellular network.

- The **SIM** (**S**ubscriber **I**dentity **M**odule) is a smart card that has a computer and memory chip permanently installed in a plastic card. It provides the mobile equipment with an identity; without it the mobile is not operational (except for emergency calls). Subscriber parameters, personal data, personal phone numbers and incoming SMSs are stored on the SIM card.

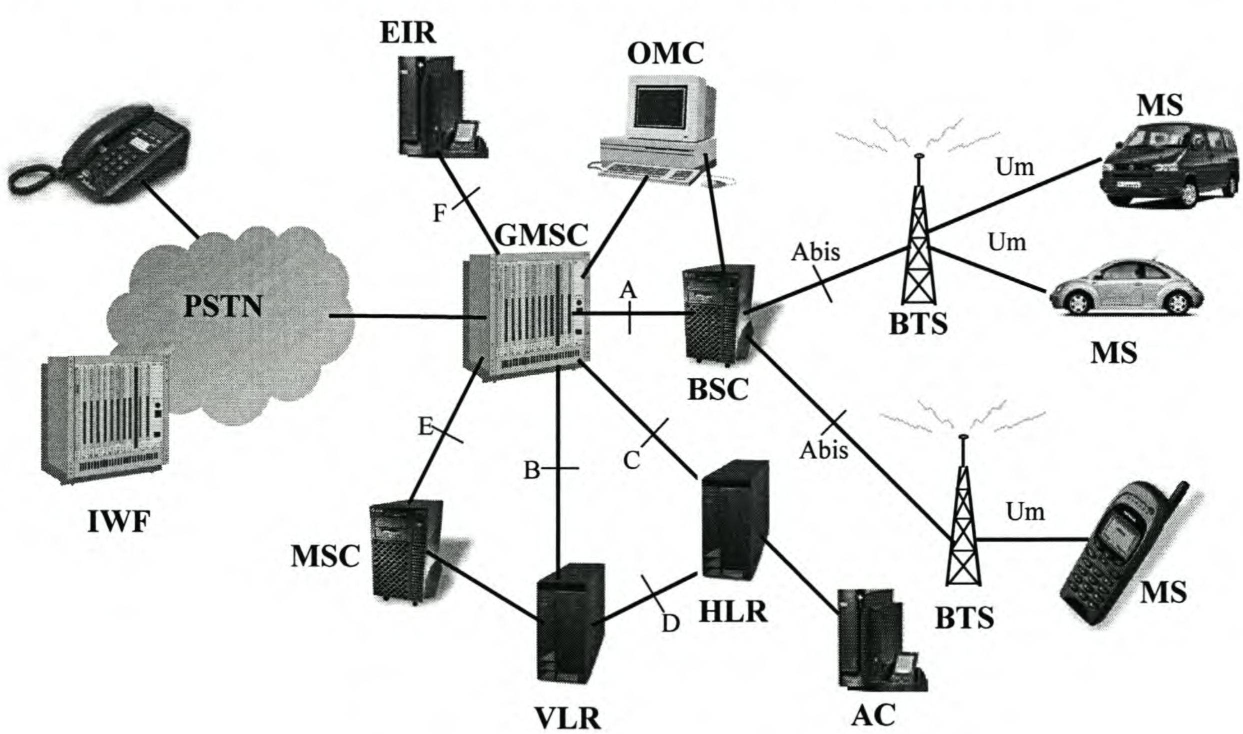

Figure 3-2 GSM System Architecture

- The **BTS** (**B**ase **T**ransceiver **S**tation), which is the counterpart to a mobile station, interfaces the mobile to the network. It is normally located in the centre of the cell where the transmitting power, on one of up to sixteen transceivers, determines the cell size.

- The **BSC** (**B**ase **S**tation **C**ontroller) monitors, controls and does frequency management of several base stations. The BSC and BTS collectively form a functional entity referred to as the **BSS** (**B**ase **S**tation **S**ubsystem).

- The **GMSC** (**G**ateway **M**obile **S**ervices switching **C**entre) is a complete exchange that interfaces the cellular network to the PSTN and provides the network with specific data about mobile stations. It is therefore capable of routing calls between the PSTN and mobile stations via the BSC and BTS.

- The **HLR** (**H**ome **L**ocation **R**egister) stores the identity, the permanent subscriber user data and temporary data. The **IMSI** (**I**nternational **M**obile **S**ubscriber **I**dentity number), the authentication key, activated supplementary services, and the phone number from the public network are permanent subscriber data. Temporary data includes the **VLR** (**V**isitor **L**ocation **R**egister) which administers the mobile station, the number to which calls are to be forwarded when activated, authentication and ciphering transient parameters.

- The **MSC** (**M**obile **S**ervices switching **C**entre) supports GSMCs by providing additional exchange capacity for traffic within the cellular network. The MSC has capabilities similar to the GMSC except that it does not access the fixed network and has no HLR.

- The **OMC** (**O**peration and **M**aintenance **C**entre) has access to the GSMC and the BSC, handles error messages from the network, and does load management of the BSC and the BTS.

- The **VLR** (**V**isitor **L**ocation **R**egister) supports the GMSC and MSC during call establishment and authentication procedures by furnishing specific subscriber data of mobiles currently served by the GMSC or MSC. The permanent data is the same as data found in the HLR. This reduces data traffic to the HLR.

- The **AC** (**A**uthentication **C**entre) assists the HLR with authentication by selecting a specific authentication algorithm for a specific subscriber, from a stored set of algorithms, by calculating the input values and by issuing the results to the HLR.

- The **EIR** (**E**quipment **I**dentity **R**egister) is an optional network security feature that stores a list of **IMEI** (**I**nternational **M**obile **E**quipment **I**dentity) numbers of mobile equipment reported stolen or those with hardware defects that may not be used in the network. The IMEI contains the mobile station's serial number, the manufacturer, the country of production and type approval.

- The **IWF** (**I**nter**W**orking **F**unction) acts as interface with other networks to manage data connections terminating outside the cellular telephone system [56]. The IWF, which in essence is a part of the MSC, provides subscribers with access to data rate and protocol conversion facilities [25]. It transmits data between a GSM DTE and a landline DTE. Communication with fixed-line PSTN telephones or modems is provided by the IWF transforming the digital voice or data frames of the GSM network to analogue waveforms. The IWF thus provides mobile devices with an end-to-end circuit abstraction, hiding from the wired network the physical details of the connection with the mobile station [56]. The IWF interfaces to PSTN, ISDN, **CSPDN** (**C**icuit-**S**witched **P**ublic **D**ata **N**etworks) and **PSPDN** (**P**acket-**S**witched **P**ublic **D**ata **N**etworks) [25].

The signalling protocol in GSM is structured in three general layers, depending on the interface. Layer 1 is the physical layer, which uses the channel structures. Layer 2 is the data link layer.

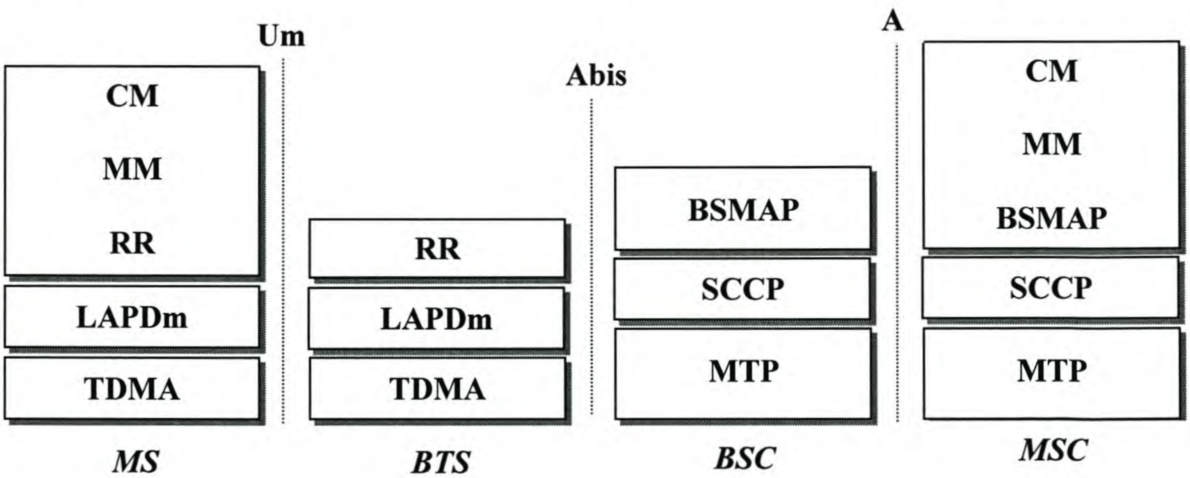

Figure 3-3 GSM Signalling Protocol Structure

Across the *Um interface*, the data link layer is a modified version of the LAPD protocol used in ISDN, called LAPDm. Across the *A interface*, the Message Transfer Part layer 2 of Signalling System Number 7 is used. Layer 3 of the GSM signalling protocol is divided into 3 sub-layers:

- **Radio Resources Management (RR)** controls the setup, maintenance, handover and termination of links, both radio and fixed, between the MS and the MSC.

- **Mobility Management (MM)** manages the location updating and registration procedures, as well as security and authentication.

- **Connection Management (CM)** handles general call control, supplementary services and SMSs.

Figure 3-4 shows the relationship between the mobile unit (as described in section 3.1), the mobile station, the base unit (as described in section 3.1), the base unit's communication equipment and the interconnecting network. The illustrated network and the base unit modem are both generic components since they may assume the role of different entities under different scenarios or circumstances.

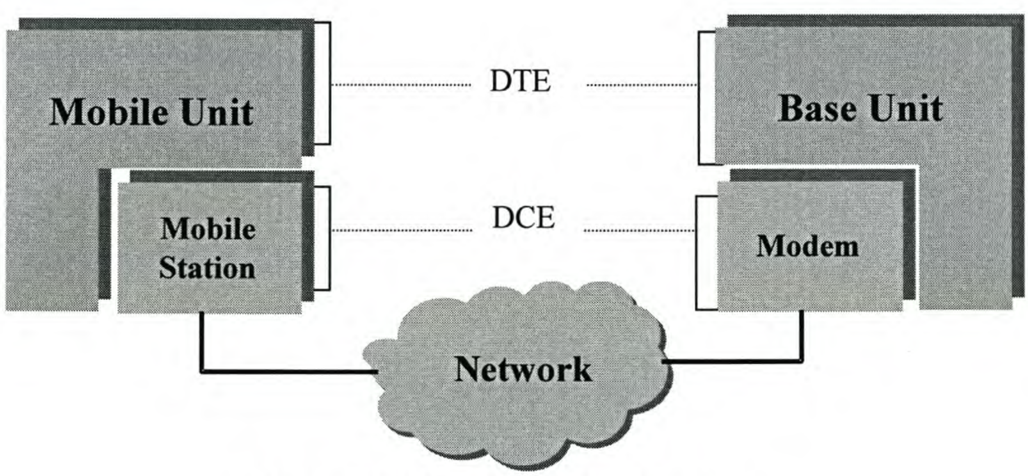

Figure 3-4 Component Architecture

The mobile station is one sub-system of the mobile unit. Functionality of the mobile unit transforms the mobile station into an *intelligent* communicating device. The base unit has a generic modem as one of its sub-systems. The modem may be a mobile station, a PSTN fixed-line modem, ISDN communicating device, etc. When the mobile station opens a data connection to a normal fixed-line telephone, the illustrated network in Figure 3-4 represents both the GSM and PSTN networks. The illustrated base unit modem is then a fixed-line PSTN modem. When the mobile station establishes a connection with another GSM subscriber, the illustrated network represents the GSM network and the illustrated base unit modem is then a mobile station. In the case where the mobile station establishes a connection with an ISDN user, the illustrated network represents both the GSM and ISDN networks and the illustrated base unit modem is an ISDN modem.

3.4 Prototype Protocol Stack

A layered protocol stack approach was chosen to enhance scalability and flexibility for future development and additions. The prototype protocol stack, called the DX Protocol Stack, is illustrated in Figure 3-5. The DX Protocol Stack consists of a device layer, a network layer and a GSMFTP layer. The fourth layer represents fleet management functionality.

The functionality for each layer is noted on the left-hand side of the diagram. This diagram illustrates the data flow between the application processes, the DX protocol stack and the physical network at the bottom.

The device driver layer directly configures, receives and transmits data over the selected communication device. The GSM driver supports SMS and circuit-switched data channel communication. According to system requirements, transparent communication links should be established via an RS232 port and via the GSM network. The top-level application layer functions the same irrespective of the communication channel used.

The DXN layer has two clients; namely, the SMS client process and the GSMFTP server. The DXN layer supports transparent communication over one of the selected communication media. It acts as multiplexer/demultiplexer to/from the lower communication devices.

The GSMFTP layer uses DXN services to do a robust file transfer to or from a specified destination. This functionality is discussed in detail in the following sections.

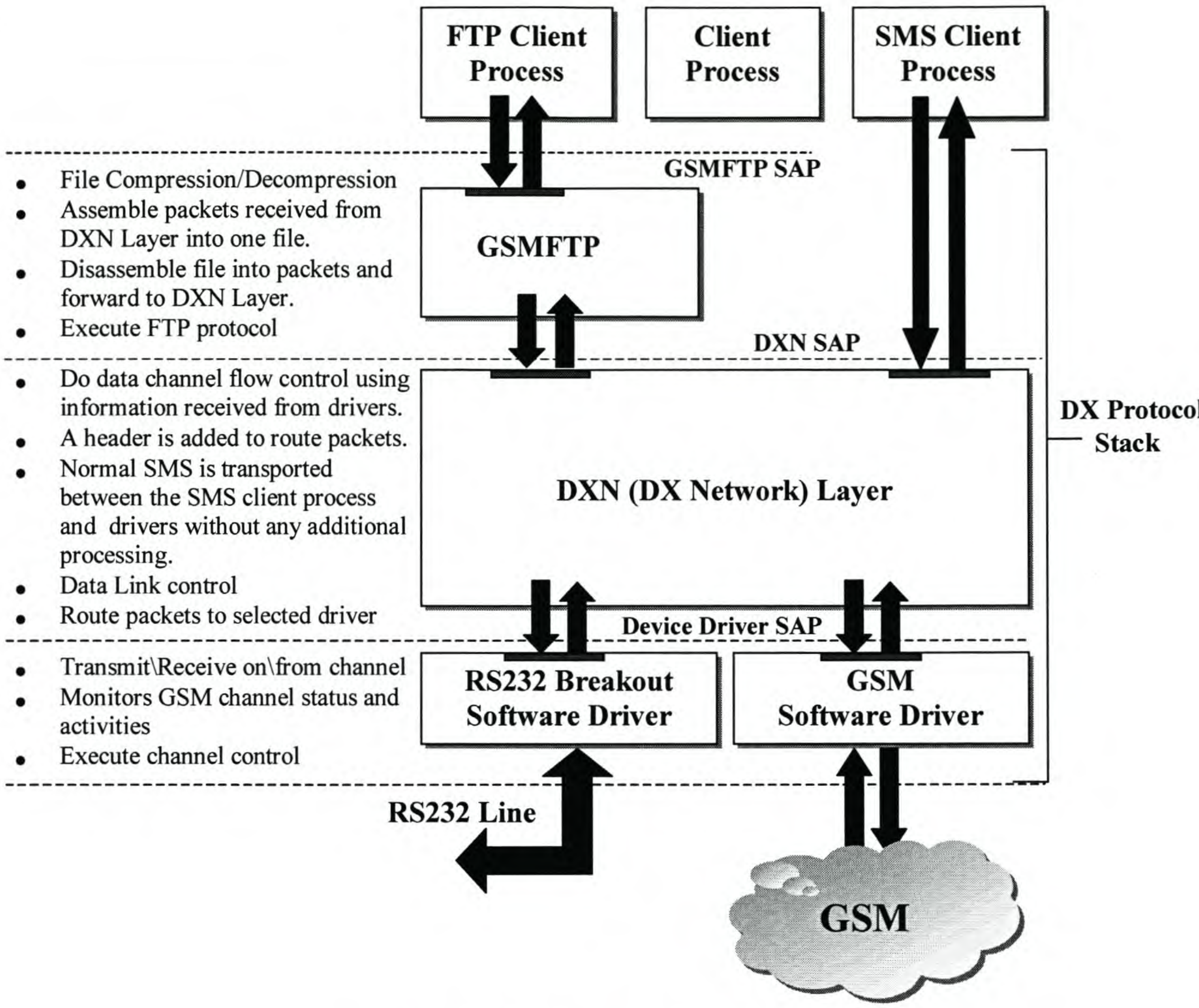

Figure 3-5 DX Protocol Stack [18]

A top-level **SDL** (**S**ystem **D**escription **L**anguage) representation of the DX protocol stack is given in Figure 3-6. The *rectangles* represent the main functional entities; namely the protocol layers. All of the layers (GSM Driver, RS232 Driver, DXN, GSMFTP and Application layers) are shown in this diagram. The arrows between these entities represent communication channels (see [13] for an SDL introduction). They may be half-duplex or full duplex. All the communication channels between the upper layers are fully duplex. These channels represent communication with external devices. The GSM driver manages communication with a GSM modem and the RS232 serial driver manages communication with DCE devices such as computers. The inscriptions or text at the channel termination points represent the set of valid signals or messages. As seen on this diagram the GSM driver can receive an *RxByte* signal from the GSM modem. The GSM modem can receive a *TxByte* signal from the GSM device driver.

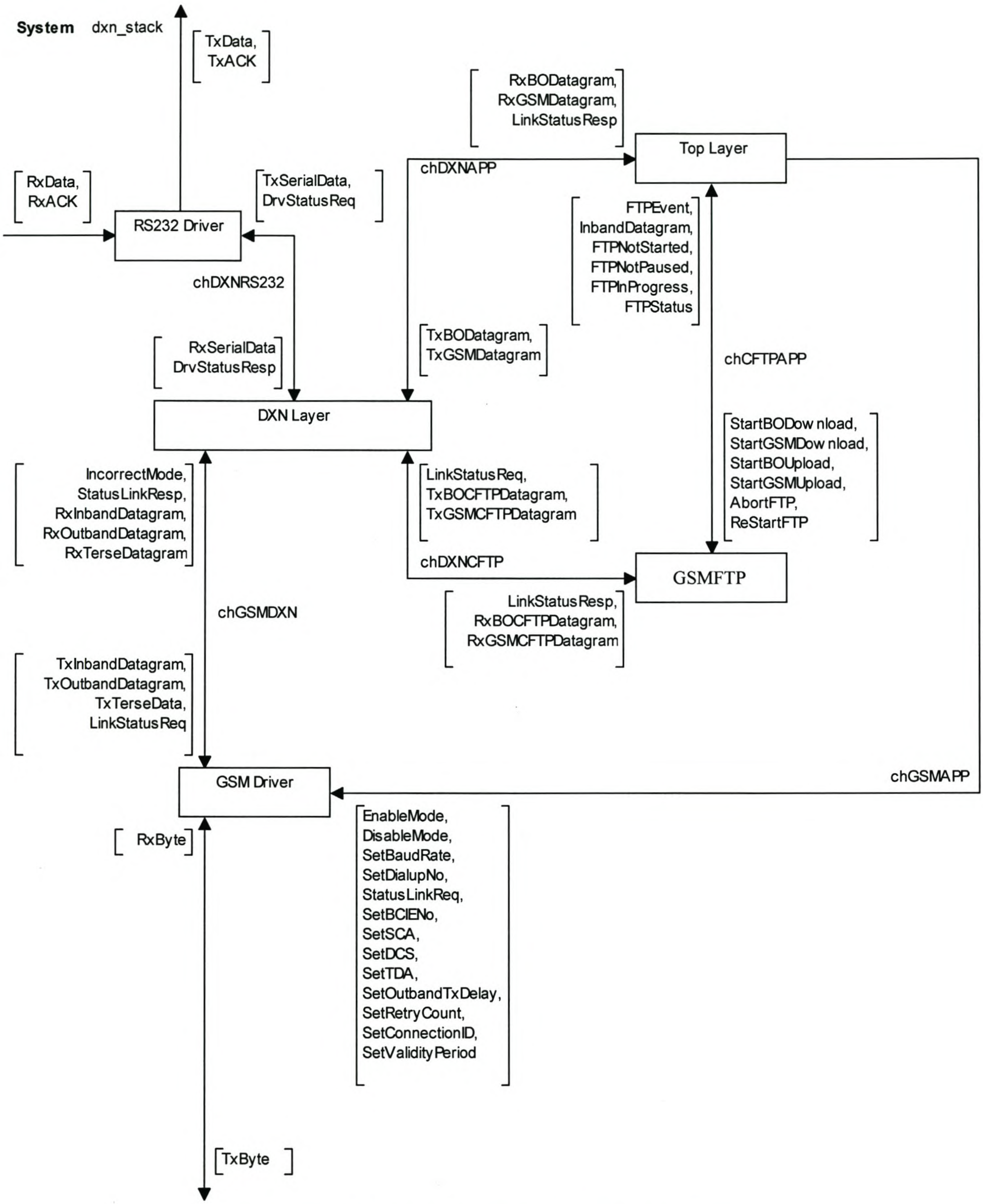

Figure 3-6 SDL System Overview of DX Protocol Stack

3.5 Service Access Points

The **SAP** (**S**ervice **A**ccess **P**oint) of every layer, indicated in Figure 3-5, is the interface through which the client process, higher in the stack, requests services from a layer lower in the stack. The SAP may be used by a lower layer to issue a message to a layer higher in the stack. The FTP client requests FTP services from the GSMFTP layer via the GSMFTP SAP. DXN services are requested via the DXN SAP and device driver services are requested via the device driver SAP.

The SAPs to all of the layers are implemented in a standard, uniform manner throughout the implementation. This reduces the effort needed to get an operational prototype protocol stack. Code re-usability is promoted: once one SAP is implemented, the code may be re-used for the other layers as well. This reduces complexity and enhances modularity

3.5.1 Mobile Unit Service Access Points

The mobile unit is implemented on an embedded platform to enhance mobility and save on power consumption. Communication implementations containing protocol stacks are examples of real-time applications. **RTXn** (**R**eal **T**ime **E**xecutive **N**ano) is a multi-tasking, non-preemptive kernel and ideal to support real-time applications [1]. RTXn is implemented on SUNSAT and proved to be a stable operating environment for moderately complex systems. Since RTXn is compact enough for implementation on a micro-controller in an embedded environment [41] with limited memory resources it is a perfect candidate to support the DX protocol stack.

Communication between layers uses standard structured messages to convey information in a familiar and recognisable manner from one layer to another. Figure 3-7 shows the structure of messages that are exchanged between any two layers.

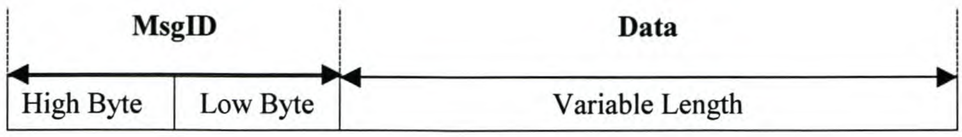

Figure 3-7 Layer-to-Layer Message Structure

The following fields are contained in the message, namely:

- **MsgID** (**M**essage **ID**entifier) is a 16-bit number that uniquely identifies the message and the corresponding data expected to be contained in it. It is stored as two octets in big endian format.

- The data field contains any number of octets.

Implementations on RTXn consist of a number of predefined tasks. These tasks may communicate with each other using a number of different mechanisms. The RTXn mailbox is one of a number of communication mechanisms. A mailbox assigned to a specific task buffers messages to that task. A message destined for a specific task is placed in the mailbox assigned to that specific task. This message remains in the mailbox until the task is scheduled. One message is removed from the mailbox when it is serviced. At the heart of the RTXn kernel is a deadline driven non-pre-emptive scheduling mechanism. RTXn attempts to satisfy the deadline requirement by scheduling the task before its deadline expires. This is a best-effort system that attempts to satisfy all of the deadlines. In the case of a mailbox, deadlines may be defined as the maximum period of time a message may be resident in a mailbox. Tasks are scheduled in the attempt to meet this deadline.

3.5.2 Base Unit Service Access Points

The base unit requires user-friendly interfaces, with flexible memory resources and processing speed to ultimately process multiple requests and multiple data transfers from many mobile units. A desktop computer with Windows 98 was selected to illustrate the prototype base unit.

TCP/IP ports are used as SAPs. Each of the different layers is assigned different, predefined but unused TCP/IP port numbers. When a service is requested from a specific layer, the message is transmitted to a specific TCP/IP port. The message handler assigned to the layer temporarily places the message into a buffer until further processing is possible (see Figure 3-8)

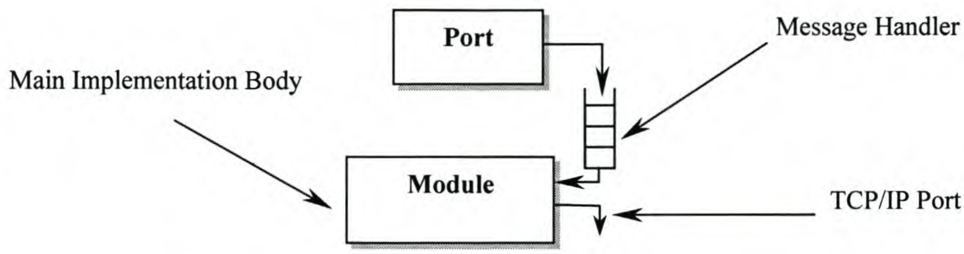

Figure 3-8 TCP/IP SAPs for Base Unit [27]

The following example in Figure 3-9 illustrates bi-directional communication between two tasks (Module A and Module B). Messages received on port A are inserted into the message queue of Module A. Messages received on port B are inserted into the message queue of Module B.

Module B connects to port A on initiation. Module A connects to port B. To send a message from Module A to Module B, the message is sent by Module A to port B. Module B receives this message on port B and places it into its queue. Once the message reaches the front of the queue it is processed by Module B's message handler. If Module B wants to acknowledge the successful processing of the received message it then sends an acknowledgement message to port A. Once Module A receives the acknowledgement on port A, it is placed into its queue. Once the acknowledgement reaches the front of queue A, it is processed.

The two SAP mechanisms on the mobile and base unit function on similar principles. TCP/IP SAPs have the added advantage that different layers of the same protocol stack may operate in a distributed fashion. Layers therefore are not bound to the same server.

Device drivers may be operated on one server to which the physical communication devices, like GSM wireless AT modems, are connected. The rest of the protocol stack may be placed on a desktop computer that may communicate over the Internet with the device driver layer via its SAP.

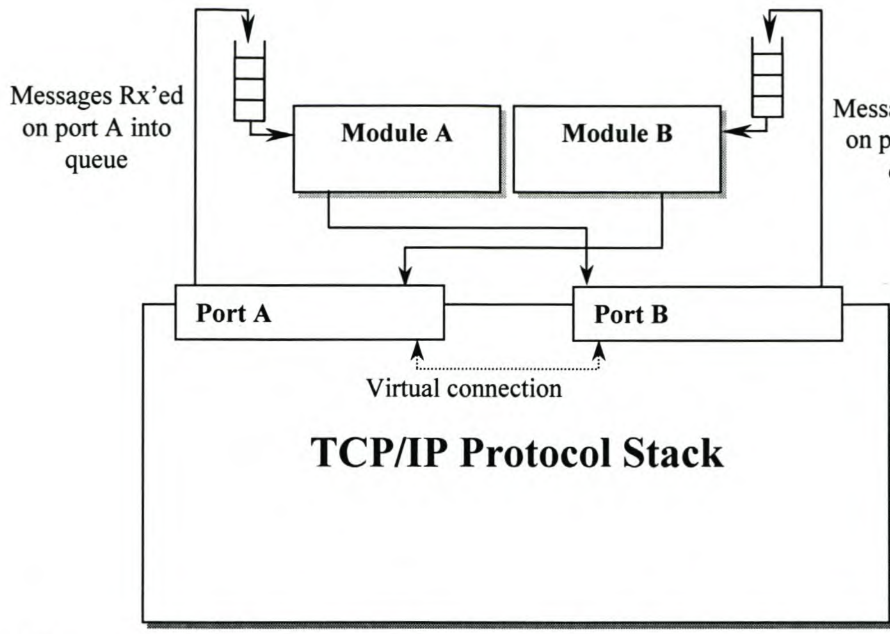

Figure 3-9 Message Communication between Module A and Module B

3.6 Device Driver Layer

The device driver layer consists of device drivers, namely

- GSM driver for controlling the GSM Wireless AT modem, and
- Serial driver for managing communication across a RS232 port.

GSM and RS232 serial driver functionality are illustrated in Figure 3-10. The DXN layer acts as client of the Device Driver Layer, requesting services such as the transmission of datagrams. Appendix A.1.1 illustrates the functionality of the GSM driver.

3.7 GSM Driver

The GSM Driver is responsible for configuring the GSM Hayes compatible modem, controlling the reception and transmission of SMSs and the establishment, control and disconnection of data channels.

The **ETS** (**E**uropean **T**elecommunications **S**tandard) GSM 07.07 specifies a profile of AT commands. It recommends that this profile be used for controlling the **ME** (**M**obile **E**quipment) functions and GSM network services from the **TE** (**T**erminal **E**quipment) through the **TA** (**T**erminal **A**dapter) [21]. GSM 07.07 assumes an abstract architecture comprising a TE (e.g. a computer) and a ME interfaced by a TA (see Figure 3-10)

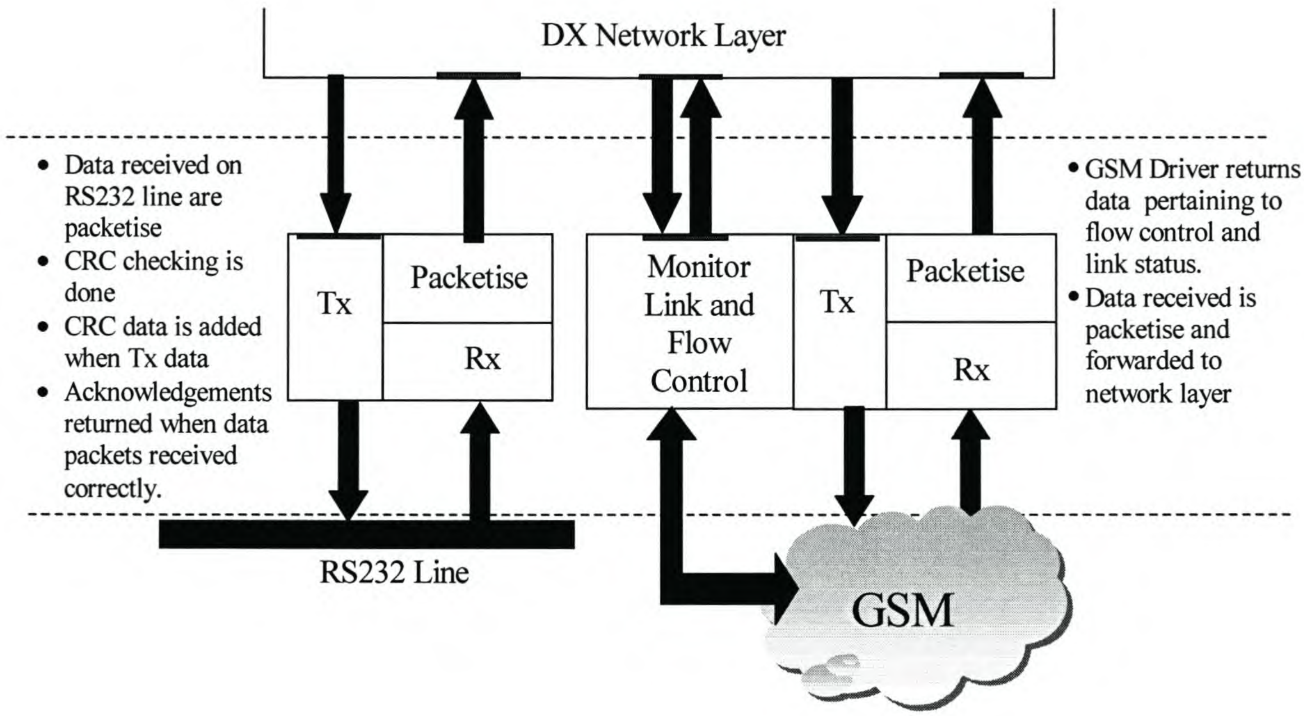

Figure 3-10 Device Driver Layer Functionality

The physical implementation of this abstract architecture may lead to [21]:

- TA,ME and TE as three separate entities;
- TA integrated under the ME cover, and the TE implemented as a separate entity;
- TA integrated under the TE cover, and the ME implemented as a separate entity;
- TA and ME integrated under the TE cover as a single entity.

Motorola's d10 integrated wireless modem was used in the mobile unit implementation. This wireless modem is a Phase 1 GSM class 4 embedded module package with voice, data, fax and SMS support [29]. It is designed to support a range of subsidiary services associated with navigation, emergency services, road tolls, security systems such as car alarm, fire alarm, etc. It is capable of supporting standard voice and data communication. The d10 is designed for use in a system environment comprising a GSM mobile radio network with one or more operators per country.

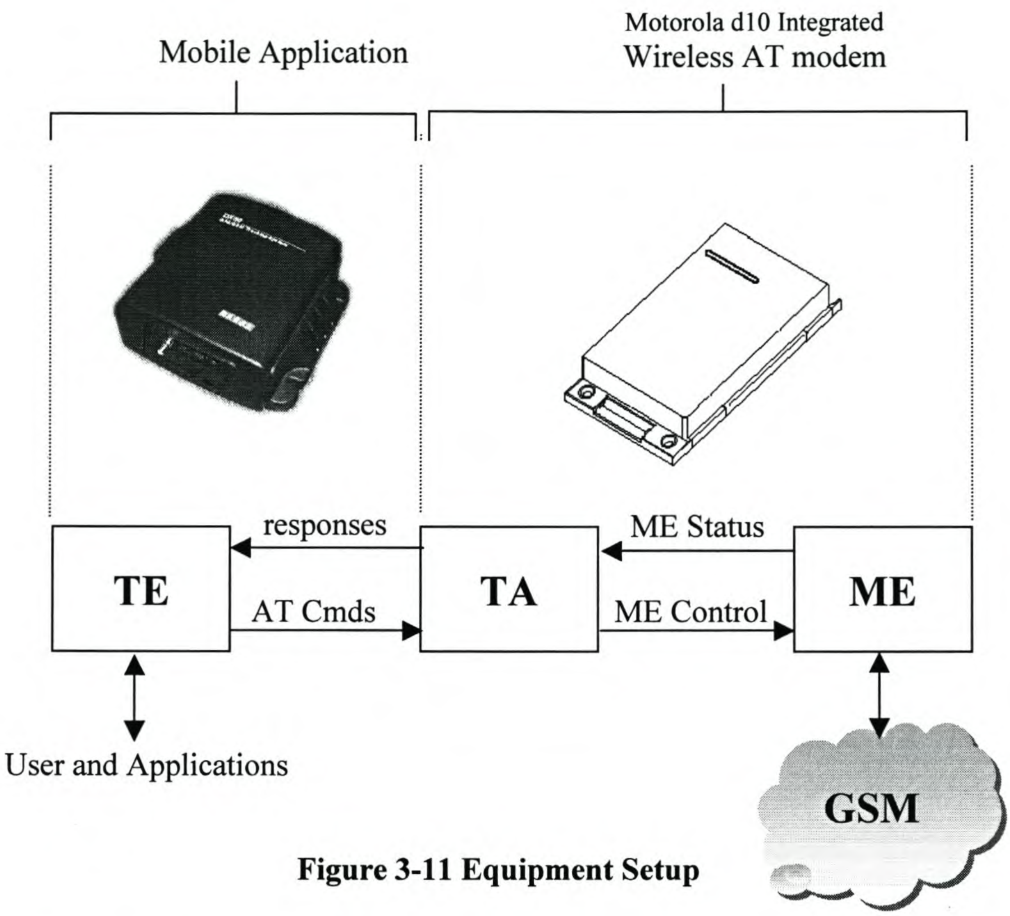

Figure 3-11 Equipment Setup

AT+C commands conforming to GSM 07.07 and GSM 07.05 and a number of manufacturer-specific AT commands are available via the serial interface of the d10 for the implementation of the functionality [29].

The interface between TE and TA is intended to operate over existing serial (ITU-T Recommendation V.24) cables, infrared link, and all links with similar behaviour. Motorola's d10 AT modem acts as a TA integrated under the cover of the ME.

3.7.1 AT Commands

Figure 3-12 illustrates the general structure of an AT command line that may be issued to AT modems. GSM commands use syntax rules of *extended* commands. Every command has a *test* command (trailing "=?") to test the existence of the command and to acquire information about the type of its subparameters [21]. Parameter type commands also have a *read* command (trailing "?") to check the current values of the subparameters.

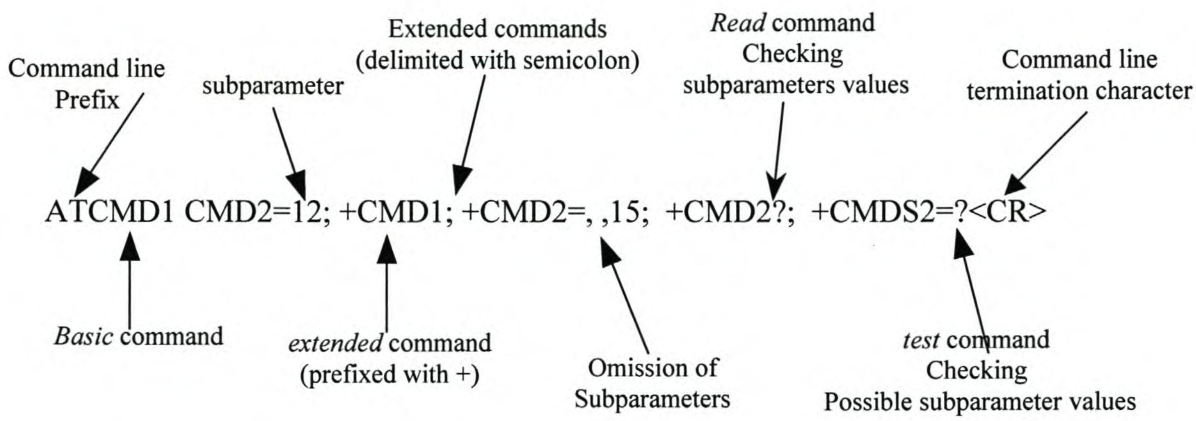

Figure 3-12 Basic Structure of a AT command line [21]

When verbose responses are enabled and all commands on a command line have been performed successfully, the result code <CR><LF>OK<CR><LF> is sent from the TA to the TE. If numeric responses are enabled, result code 0<CR> is sent instead.

3.7.2 SMS Mode

The GSM Technical Specification defines three interface protocols to control SMS functions within a GSM mobile telephone from a remote terminal via an asynchronous interface. These three interfaces are [20]:

- **Text Mode** is a character-based interface based on AT commands and is suitable for unintelligent terminals and terminal emulators.

- **Block Mode** is an 8-bit binary protocol with error detection and is suitable to control remote devices via unreliable links. The Block Mode Protocol encapsulates the SMS **PDU** (**P**rotocol **D**ata **U**nit) used for short message transfer between the mobile station and the SMSC.

- **PDU Mode** is a character-based protocol interface providing control of the protocol elements contained within the SMS PDUs. This mode is suitable for software drivers based on AT command structures which do not understand the content of the message blocks and can only pass them between the MT and upper level software resident in the TE.

Block mode is an older and less efficient operating mode for SMS communications. The PDU mode, which is much easier to work with, is recommended for SMS communication[29]. PDU packets have a fairly complex structure, so careful examination of GSM 04.11 and GSM 03.40 is required. The d10 wireless modem uses PDU mode to transmit and receive SMSs.

Several AT commands are used to control the wireless GSM modem. The GSM device driver can execute a subset of AT commands and can recognise a number of standard responses. These commands are responsible for configuring, control and break-up of data circuit calls, sending SMSs, receiving SMSs and deletion of SMSs stored on the SIM card

The GSM driver accepts verbose responses from the AT modem. The responses that can be recognised by the device driver are OK, ERROR, new SMS message indication, incoming data call, no carrier, reset, data channel connect, state of phone and list of stored SMSs (see Figure A.7 and Figure A.8)

.

3.7.2.1 PDU frame

The format of the PDU frame is illustrated in Figure 3-13. This structure is not too complex but is powerful enough to configure for use and transmission of data to various destinations, using various types of carrier networks and various termination equipment. A detailed description of the PDU structure may be found in [29] and [20]. The PDU is a datagram issued to the GSM modem containing enough information to route data from its source to its destination containing information such as the SCA, Coding schems, destination, destination type, etc.

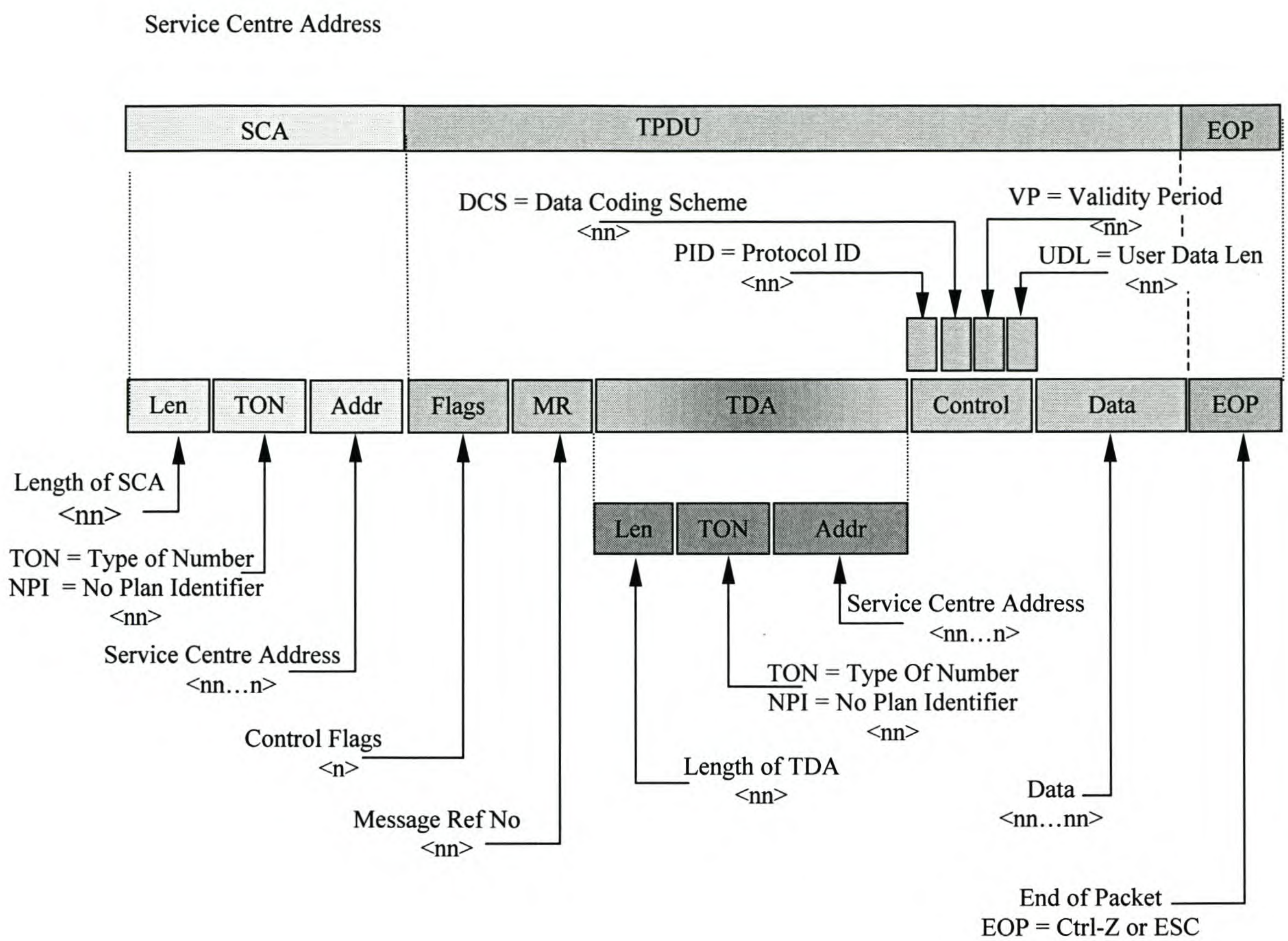

Figure 3-13 SMS PDU Structure

3.7.2.2 Inbound SMSs

New SMSs are stored on the SIM card until they are deleted under instruction from the TE. A standard message, indicating the arrival of new SMSs, is issued to the TE. All stored SMSs may be listed. Each of the listed SMSs stored on the SIM card generates a separate message (*TxOutbandDatagram*, Figure A.1) that is issued to the DXN layer. Only the SMS data is encapsulated in the message issued to the DXN layer.

3.7.2.3 Outbound SMSs

All outbound SMSs are stored in a round-robin FIFO queue named the Out-Of-Coverage queue. Outbound SMS are stored in this queue until the mobile unit moves into GSM coverage again.

As long as there are messages in the queue, the device driver will periodically attempt submission of SMSs until successful.

3.7.3 Terse Mode

In Terse mode data from the device driver client is transmitted verbatim. Any data received on the GSM port while the driver is in this operational mode is forwarded to the DXN layer as received, without any processing done. This mode may be used for software debugging by monitoring incoming data from the AT modem byte by byte (see Figure A. 4).

3.7.4 Data Mode

From its conception in the mid-1980s, the GSM network standard was designed to allow computer data to be exchanged at a maximum speed of 9600 bps. GSM offers two modes for data transmission: transparent mode and non-transparent mode. The modes differ in their approach to data flow through the network.

- **Transparent mode** effectively offers an open pipe for the data, where the control is handled only by the transmitting and receiving modems with no intervention from the network. Transparent mode allows the use of compression to produce higher data throughput and reduce call charges. This mode is used when the modem is stationary or when travelling within a good-reception area.

- **Non-transparent mode** offers a robust data communication link up to a maximum data speed of 9600 bps using a sophisticated error-correcting protocol called **RLP** (**R**adio **L**ink **P**rotocol). It is used when the modem is moving or in areas with bad service reception.

GSM exhibits modest transmission bandwidths, has small frame sizes, and uses circuit mode operations due to the voice-orientated design. Instead of transmitting digitized voice, the data services allow the transmission of data handed to the link layer. For optimal voice performance, cellular telephone systems use short frames that suffer losses of one to two percent [56]. As long as the physical layer manages to randomize these losses, audible voice-quality degradation is avoided. The randomisation of losses or errors can be tolerated to a certain extent for voice applications but for data applications it cannot be tolerated under any circumstances.

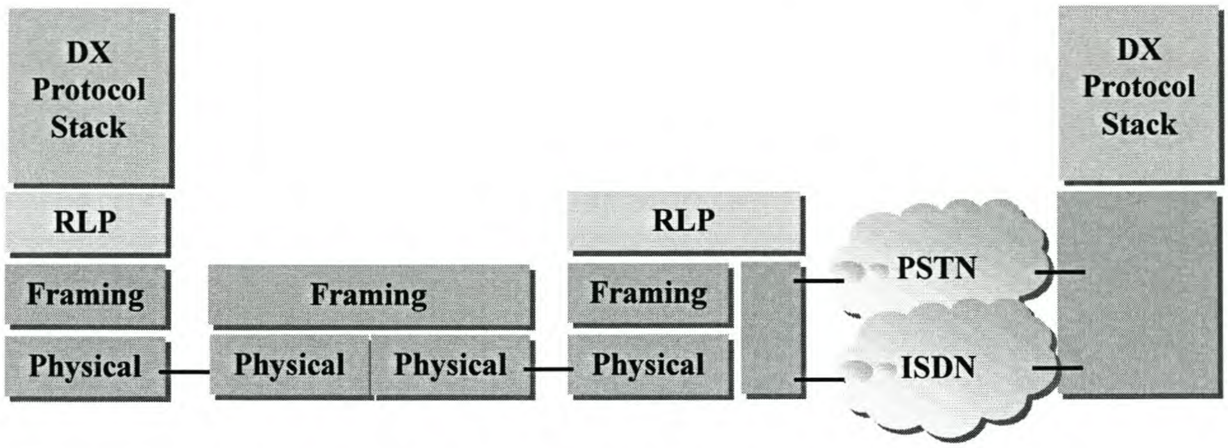

Figure 3-14 DX Protocol Stack on RLP

The IWF transforms the digital voice or data frames to analogue waveforms when a connection is established with analogue telephones or modems. To communicate with ISDN systems, the IWF performs rate adaptation and frame conversions. The IWF provides mobile stations with an end-to-end circuit abstraction, hiding from the wired network the physical details of the connection with the GSM mobile station. In non-transparent mode RLP is used between the mobile station and the IWF to enhance link functionality and performance. RLP can also perform error recovery over the cellular network to hide losses from the wired network by using for example an **ARQ** (**A**utomatic **R**epeat re**Q**uest) scheme to retransmit lost packets. Besides improving reliability it encapsulates data from the higher layers into link frames.

3.7.4.1 Protocol Encapsulation

A PDU frame, illustrated in Figure 3-15, encapsulates data on the circuit-switched data connection. All data transmitted is terminated by an END byte. The structure of the datagram is shown in the following figure.

Session ID	Type	Data	END

Figure 3-15 PDU for data on Data Channel

The 16-bit session identifier relates data pertaining to a specific session. This ultimately allows the base unit to process multiple connections/sessions simultaneously. The *Session ID* data may therefore be used for routing purposes as well.

The 8-bit *Type* distinguishes between FTP data and SMS management data. When in SMS Mode, the GSM driver will transmit remote management data via the SMS facility. When in RLP mode, messages are sent via the circuit-switched data connection. The type is used to issue the *RxInbandDatagram* for GSMFTP data and *RxOutbandDatagram* for SMS Data.

A byte-stuffing algorithm ensures that the END byte is unique. Byte stuffing ensures that occurrences of END bytes in the datagram does not prematurely indicate a false termination. All occurrences of END in the datagram are therefore replaced by a ESC_END combination of bytes (see Figure A. 6)

3.9 DX Network Layer

The network layer provides services to the GSMFTP layer and is a client of the device driver layer. The DXN Layer accepts responses from the GSM device driver when in SMS or RLP Data modes (see Figure A.12). The DXN Layer does not use the device driver terse mode. The terse mode may be used at a later stage to enable the application layer to issue valid AT commands to the terminal equipment, not implemented in the GSM driver. This is a quick way of adding AT commands to the implemented set, without changing low-level driver software.

On requesting service from the DXN Layer the client process must specify to which driver the packets must be routed. Packets forwarded to the GSMFTP or SMS client processes include enough information to indicate the origin channel (GSM or RS232) of the datagram (see Figure A.13).

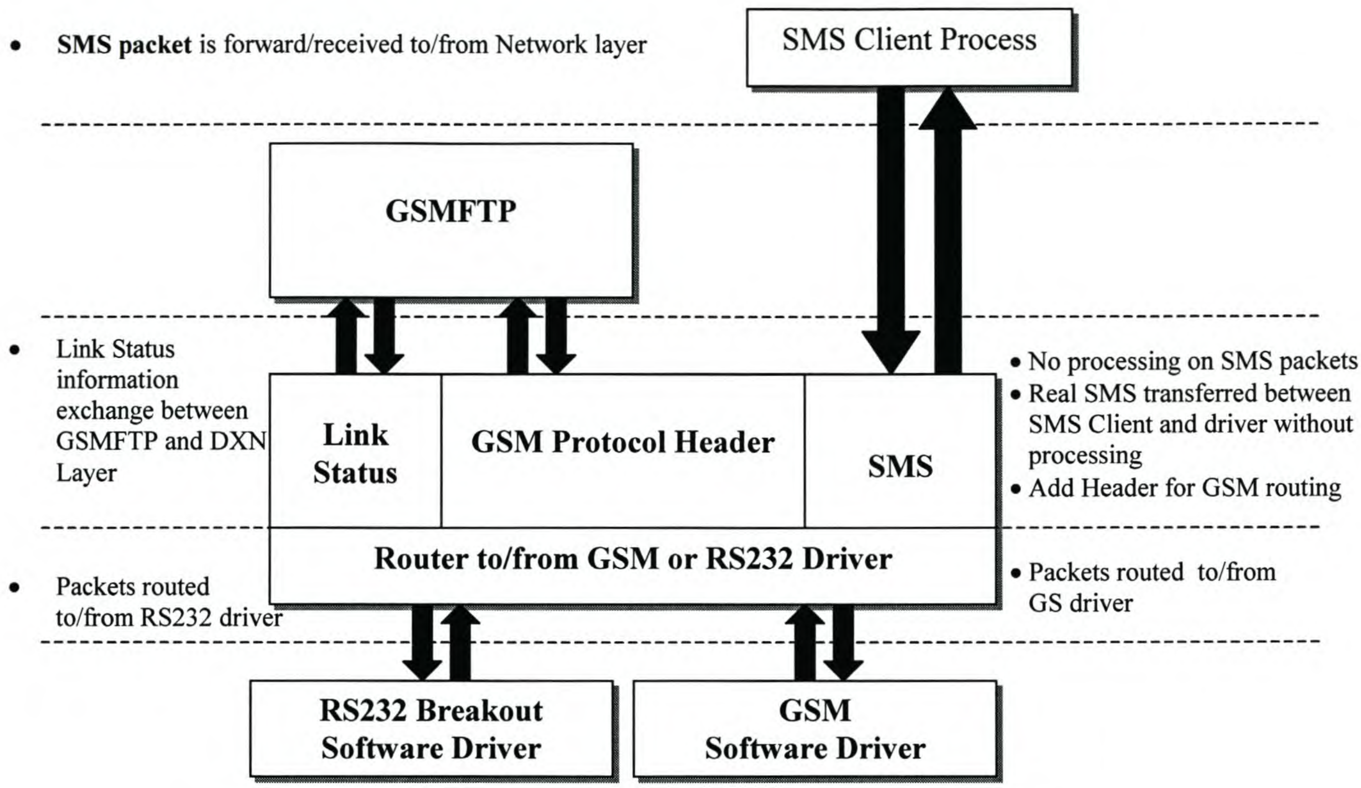

Figure 3-16 DX Network Layer Functionality

3.10 GSMFTP Layer

The GSMFTP Layer reliably transports a file between the mobile unit and base unit. The FTP Protocol engine is designed to be robust under conditions of network congestion, bad throughput, moving mobile units and terminated data connections. Stream compression and encryption may be executed before the actual file transfer starts. These compression and encryption processes are not discussed further. The layered FTP functionality is illustrated in Figure 3-17 (see Figure A.14 for SDL diagram).

Two different operations are supported by GSMFTP; namely, file uploads and downloads. The term upload is used as seen from the mobile unit's perspective. The mobile receives an upload request when data are to be transferred from the base unit to the mobile unit. This will typically be used when a firmware upgrade is to be executed

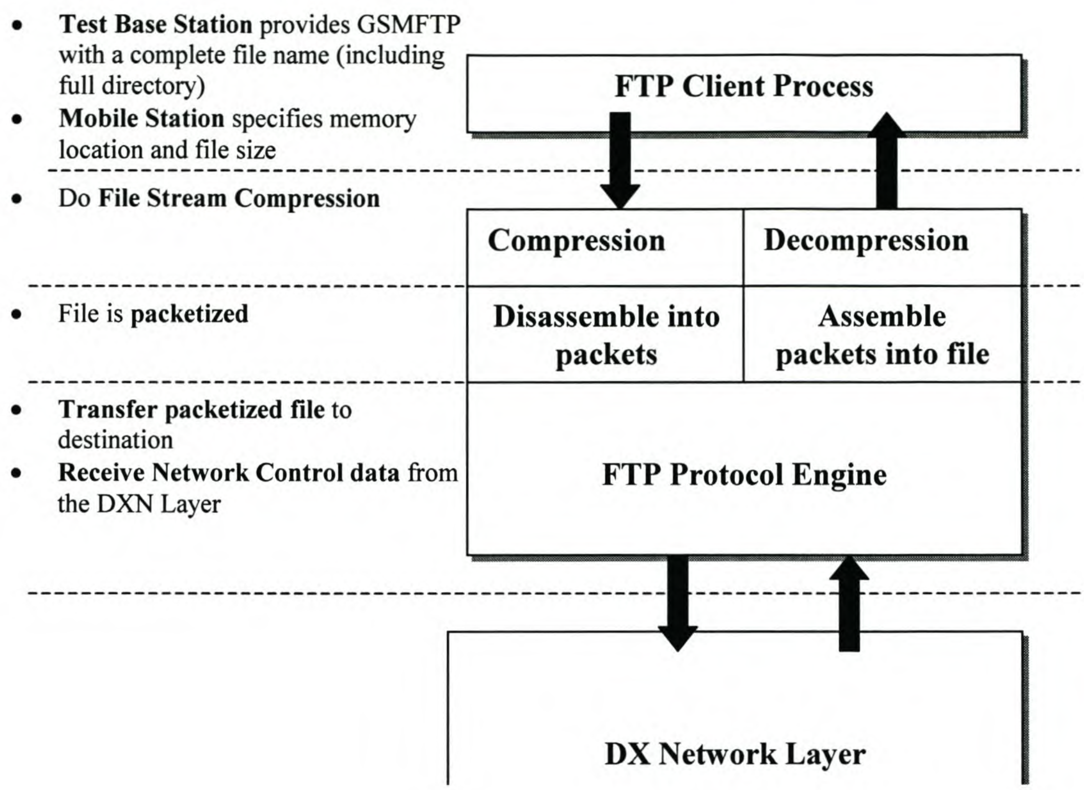

Figure 3-17 GSMFTP Functionality

3.10.1 Data Upload Transfer Procedure

Figure 3-18 is the state transition diagram for data file uploads. The FTP Protocol engines for the mobile unit and base unit go through the following states:

3.10.1.1 Request for Upload

The mobile unit issues a ***request for upload*** and waits for a response from the base unit. The base unit receives an ***upload indication***, and returns an ***upload confirm response*** when it is ready to start the GSMFTP process. This response includes the file size to be transferred between the two parties (see Appendix A.1.3.1 for SDL illustration).

3.10.1.2 Transmit Data Request

The base unit immediately starts to ***transmit data requests*** containing the data of the transferred file. Since datagrams are delivered in sequence by RLP, the ***confirm upload response*** is first received by the mobile unit containing the file size. The base unit can therefore calculate the exact number of data indications to expect. The upload confirmation response is followed by a stream of ***data indications*** (see Figure A.19, Figure A. 20 in Appendix A.13.1).

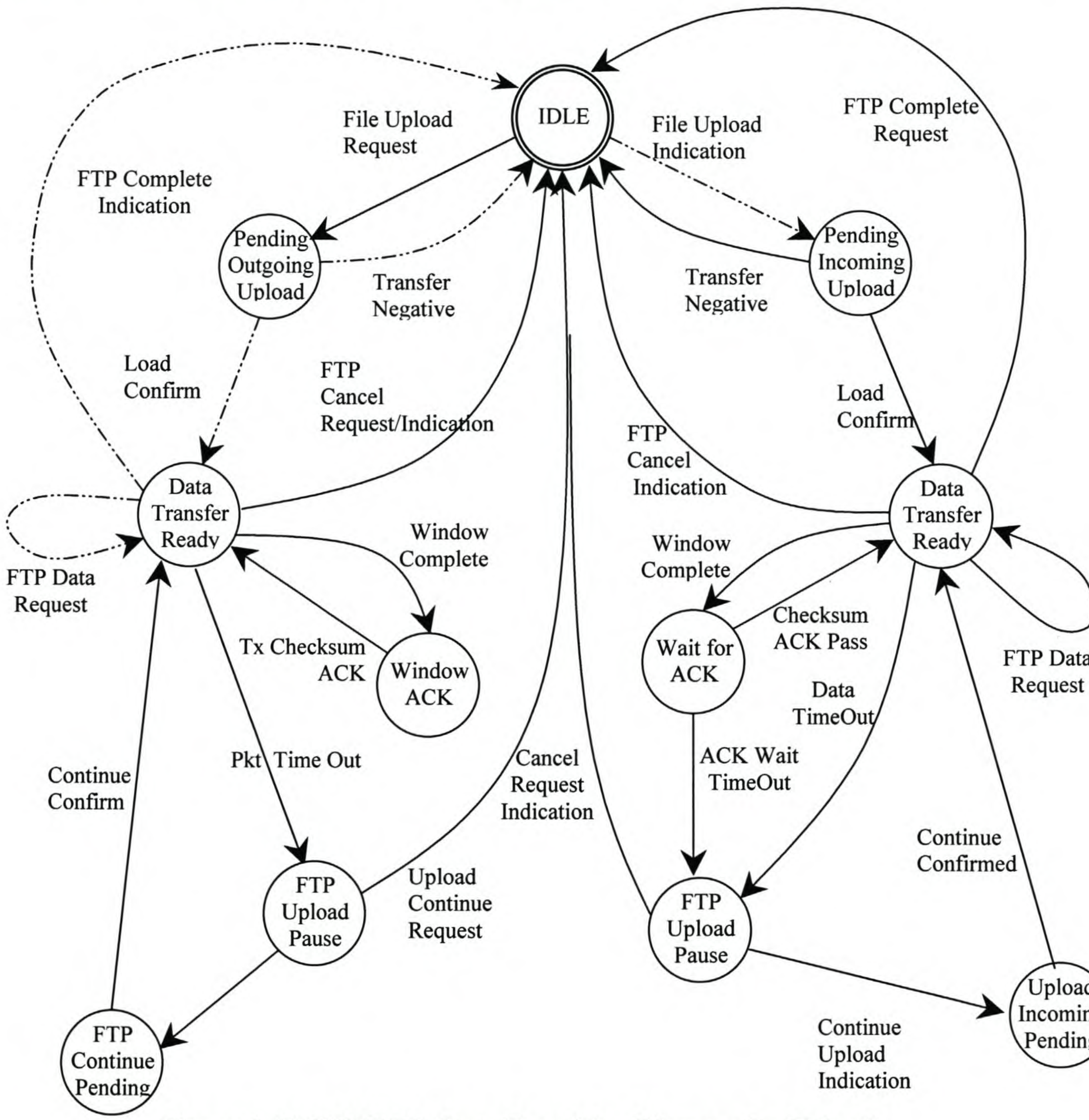

Figure 3-18 GSMFTP State Transition Diagram for Uploads

Datagrams are transmitted using a WINDOW acknowledgement scheme. Individual datagrams are not acknowledged but only a fixed number of datagrams is acknowledged at a time. These collections form what is called a window. When doing data uploads the base unit transmits a complete window and waits for an acknowledgement from the mobile unit indicating correctly delivered data (see Figure A. 24).

Once the base unit receives the acknowledgement, it starts to transmit the next window of datagrams. If the base unit does not receive an acknowledgement, it moves to the *PAUSE* state. The mobile unit only issues an *ACKnowledgement* to the base unit if the checksums match. If the checksum does not match, the mobile unit and base unit move into the *PAUSE* state. When the FTP process is CONTINUED the base unit restarts at the beginning of the last UNACKNOWLEDGED window (see Figure A.26).

3.10.1.3 FTP Complete Request

When the final data indication is received the mobile unit issues an ***FTP complete request*** and immediately moves into idle mode again. The base unit receives the ***FTP complete indication*** and moves into the idle state as well (see Figure A. 20).

3.10.1.4 Link Failure and Re-establishment

The mobile unit assumes the link has failed when no datagrams arrive within the assigned waiting period. If the mobile FTP engine is in DATA TRANSFER READY state or WAIT ACK state and the waiting time expires, then the FTP engine moves to the UPLOAD PAUSE state.

It is then up to the application on the mobile unit, to re-establish a connection with the base unit or to cancel the upload after a predetermined period has expired. As soon as a data link has been established again between the base unit and the mobile unit the events, described in the following section (3.10.1.5) , occur.

3.10.1.5 FTP Continue Request

An ***FTP Continue Request*** is issued by the mobile unit (see Figure A.26). This request contains a datagram sequence number, indicating the number of valid data indications received. This number is a multiple of the fixed window size. The base unit receives the ***FTP Continue Indication*** and re-positions the data file pointer from where data are to be re-transmitted. Sufficient information is included within the Continue Indication to re-position these file pointers.

3.10.2 Data Download Transfer Procedure

With a file download the data is transferred from the mobile unit to the base unit. Figure 3-19 shows the state transition diagram for the file downloads. This transition diagram looks the same as the state transition diagram for data uploads, apart from a small number of differences. The only differences are that:

- When a download is initiated by the mobile unit, the ***Download Request*** contains the size of the file to be downloaded. The mobile unit then awaits download confirmation (see Figure A.16).

- The base unit receives the download indication with the relevant information. If it is ready to continue it issues a ***download confirmation response*** (see Figure A.15).

- On receiving the download confirmation response, the mobile unit will start *transmitting* ***the data requests*** (see Figure A.17).

The other steps are exactly the same as for the upload file transfer.

The mobile unit assumes that the data link has failed if it does not receive an acknowledgement from the base unit when a WINDOW frame is complete. The mobile unit will enter the *DOWNLOAD PAUSE* state when the assigned waiting period has expired (see Figure A. 23). On re-establishment of the data link, the mobile unit sends a *DOWNLOAD CONTINUE* request (see Figure A.25).

The base unit issues an ***FTP Continue Response*** and start to transmit data from the requested file position. When the final data indication is received the mobile unit issues an *FTP complete request* and immediately goes into idle mode again. The base unit receives the ***FTP complete indication*** and moves into the idle state as well.

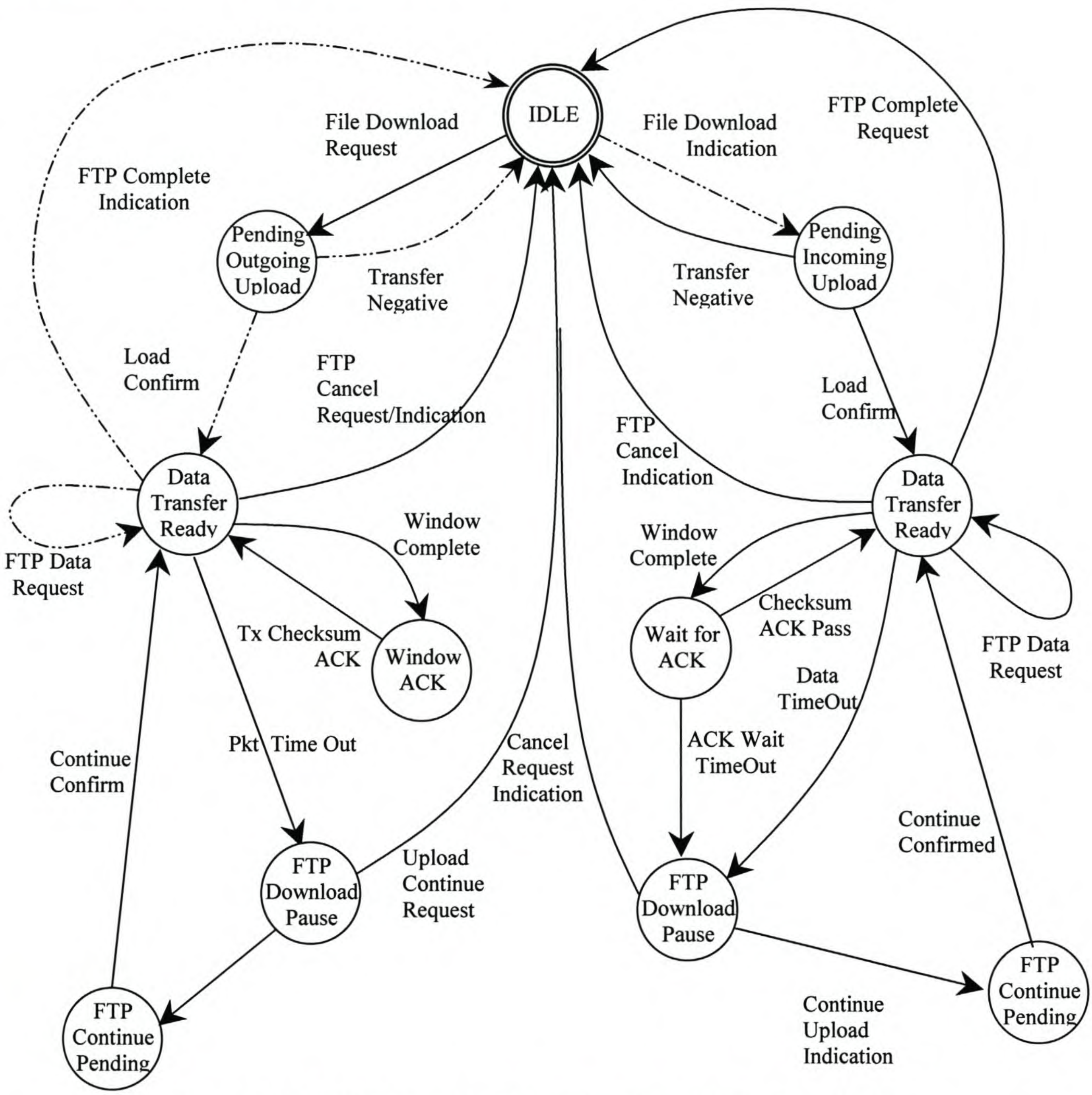

Figure 3-19 GSMFTP State Transition Diagram for Downloads

When the base unit receives the download continue indication, it returns a download continue response. Included in this response is the number of data indications already successfully received. The mobile unit must therefore re-position itself and start transmitting data from the file not yet received by the mobile unit.

3.11 GSMFTP Performance

RLP employs ARQ, forward error correction and flow control. Flow control and retransmission of faulty data blocks result in improved performance in the mobile segment of the data connection but at the expense of variations in throughput and delay [15]. The GSMFTP implements a variation of the **GBN (Go-Back-N)** error correction protocol on top of RLP. There should be no need to implement another error correction protocol on top of RLP, because the probability of error for a non-transparent RLP channel is 10^{-16} [29]. The window method was primarily introduced to act as a high-level flow control mechanism. This scheme allows the client and server to synchronise with each other once a transmission window is completed. The server should therefore never overflow the client. Once the server is finished with one transmission window it waits for an acknowledgement from the client. The client sends the acknowledgement when it has processed all of the data for that specific transmission window. If the data was correctly received and the client is ready to process the next transmission window, only then is the acknowledgement issued to the acting server.

Secondarily the GBN window scheme acts as an error correction mechanism. It could happen that interrupts, announcing arrival of data bytes, may be missed under extreme stress conditions. This is especially likely since several fleet management processes are active along with the FTP procedure. The result is a busy scheduler and processor. This GBN window method addresses both problems.

This dual functionality of the transmission window scheme is perfect for doing error and flow control at the same time. A window method, which allows the sender to have multiple unacknowledged packets, is also more efficient than a scheme where every packet is acknowledged. In general, the efficiency for GBN is defined as [15]

$$\eta_{GBN} = \frac{1}{1 + \frac{\rho}{1-\rho} N}$$

Where η_{GBN} is the efficiency of the GBN protocol, ρ is the probability of error, N is the window size

The efficiency of GBN is illustrated in Figure 3-20. Efficiency decreases rapidly with the increase in error probability. GBN is therefore effective for small error probabilities and small window sizes.

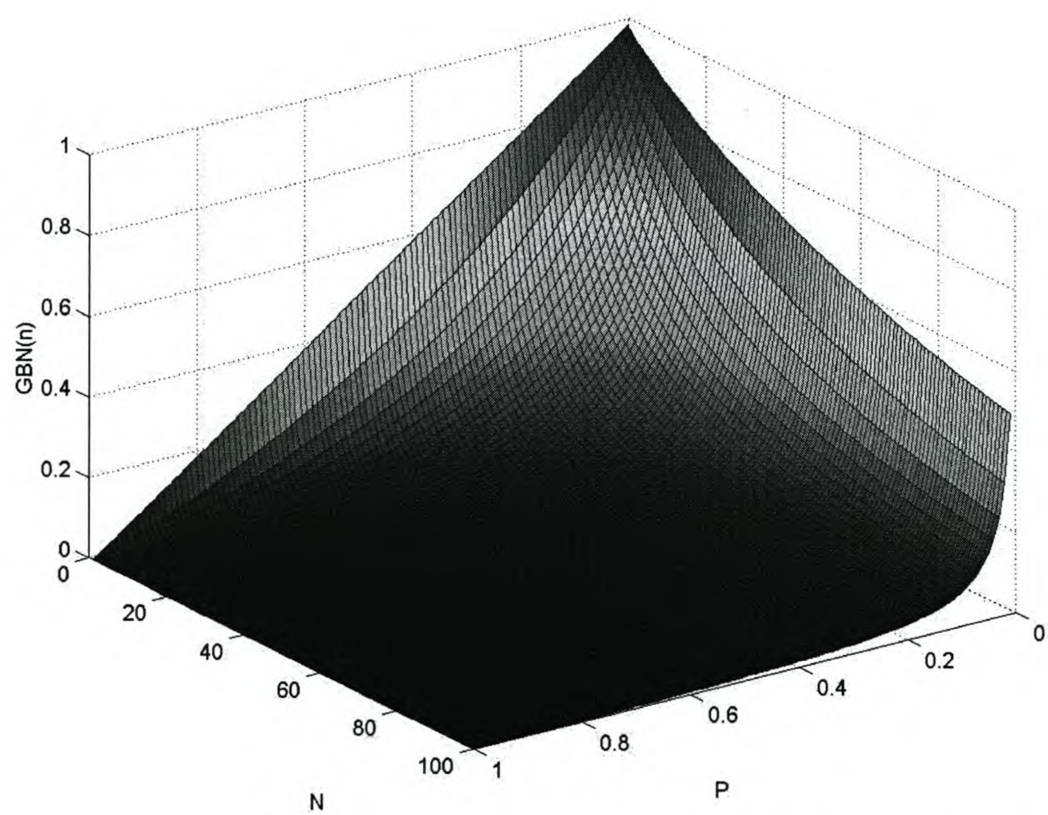

Figure 3-20 GBN Efficiency vs Error propability (P) vs Window Size (N)

SRP (**S**elective **R**epeat **P**rotocol) requires buffering at both the sender and the receiver and is more efficient than the GBN protocol. This protocol can request the retransmission of specific packets. The general efficiency of a SRP scheme is defined as [15]

$$\eta_{SRP} = \frac{2 + \rho(N - 1)}{2 + \rho(3N - 1)}$$

The efficiency graph of the SRP error recovery scheme is shown in Figure 3-21. The overall trend looks similar to Figure 3-20 except for some minor differences. The fall in efficiency on the axis (N = 0 axis, P = 0 Axis) seems steeper.

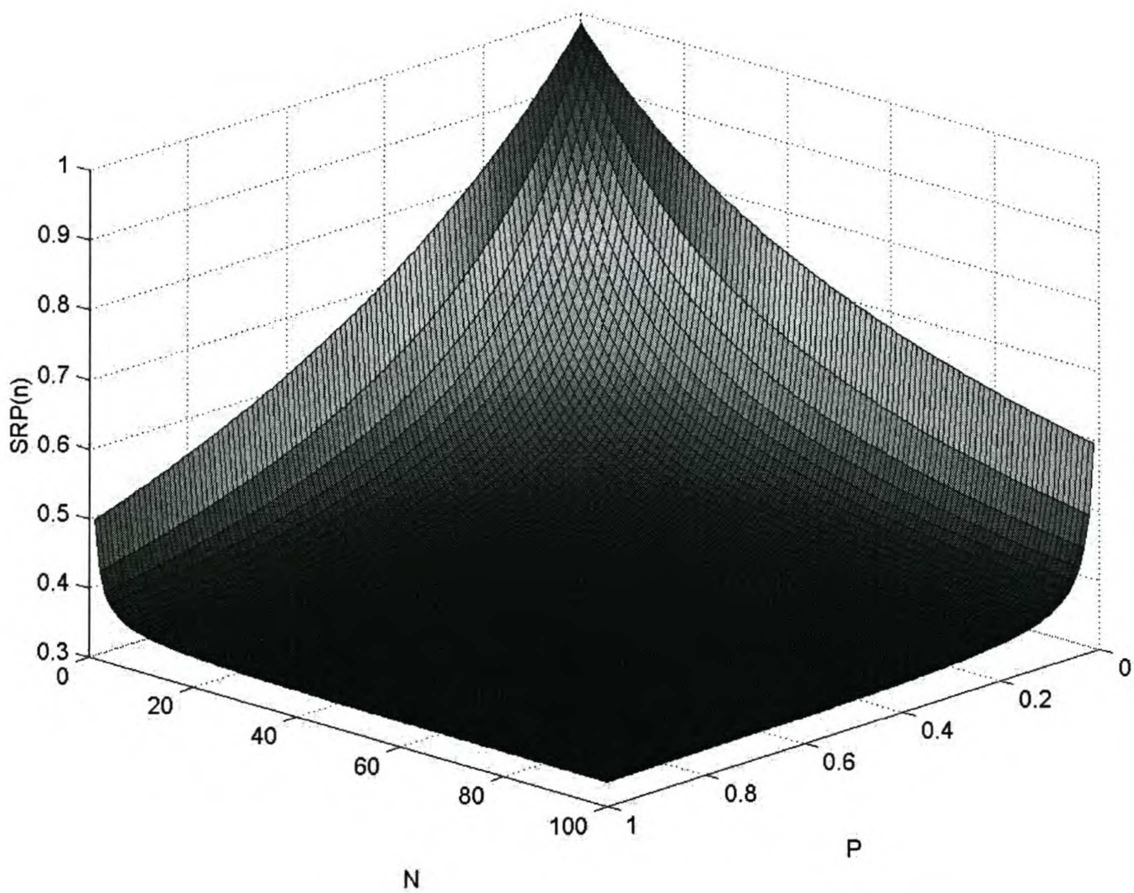

Figure 3-21 SRP Efficiency vs Error Probability (P) vs Window Size (N)

To truly see the difference in efficiency performance between GBN and SRP the efficiencies as calculated in their respective equations will be subtracted from each other Figure 3-22 shows the difference in efficiency between SRP and GBN.

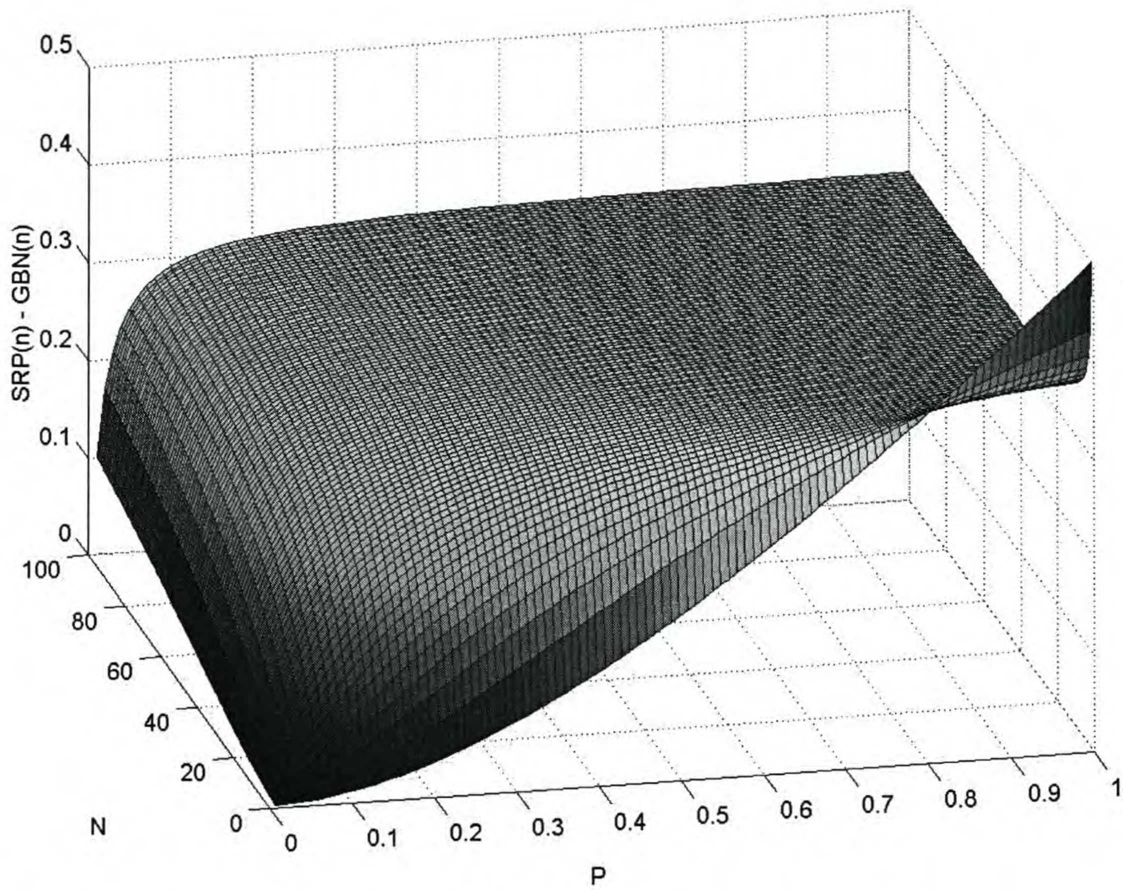

Figure 3-22 SRP - GBN Efficiency vs Error Probability vs Window Size

The efficiency on the graph is positive throughout the graph. SRP delivers better efficiency results than GBN. These two techniques deliver similar results when the probability of error is small or close to zero. SRP seems to deliver significantly better results as the probability of error and window size increases. The difference in efficiency flattens with an increase in error probability and window size.

The SRP requires buffering at both sender and receiver. SRP is more complex than the GBN scheme. With SRP only the faulty PDUs are retransmitted, but with GBN the complete window is retransmitted.

When very low error priorities are experienced, GBN may be the chosen above SRP. The reason is that there is very little difference in efficiency performance at low error priorities but GBN is a simpler implementation requiring less resources.

The previous equations do not distinguish between packet control information and user data. When a file is streamed across the data channel without any control information the channel may be classified as fully utilised. The use of a PDU structure (illustrated in Section 3.7.4) with control information and the re-transmission of a corrupted window, results in efficiencies of less than 100%. The transmission efficiency of a single PDU with 4 bytes of control information is defined as

$$\eta_{GSMFTPonepacket} = \frac{D_n}{D_n + 4}$$

Where D_n is the number of user data bytes contained in one PDU

4 is the number of control information bytes.

The efficiency of the GSMFTP Window Scheme is defined as (see Appendix B for derivation)

$$GSMFTP_\eta = \frac{ND_n(1 - P_p N)}{N(4 + D_n + T_a B) + T_{ack} B}$$

N	*fixed number of packets a window consists of;*
D_n	*number of user bytes per packet;*
P_p	*propability of packet error occurring;*
T_a	*time in seconds between successive packets in a window;*
B	*bandwidth in bits per second;*
T_{ack}	*time in seconds for window ACK to arrive;*

This equation may be optimized in terms of N by differentiating *GSMFTP* with respect to *N*. This optimized efficiency may be expressed as follows:

$$\frac{d}{dN}GSMFTP_{\eta} = \frac{D_{\eta}(T_{ack}B - 2P_{p}T_{ack}BN - P_{p}(4 + D_{n} + T_{a}B)N^{2})}{(N(4 + D_{n} + T_{a}B) + T_{ack}B)^{2}} = 0$$

The equation above will be satisfied when the numerator is zero. The numerator is a quadratic equation that has the following two null points:

$$N = -\frac{b \pm \sqrt{b^{2} - 4ac}}{2a}$$

where $a = -P_{p}(4 + D_{n} + T_{a}B)$,

$b = -2P_{p}NT_{ack}B$, and

$c = T_{ack}B$

3.12 Conclusion

The DX Prototype Protocol Stack was successfully implemented for use in a fleet management system. This protocol stack supports;

- Bi-directional communication between the mobile unit and the central base unit; and
- the base unit may communicate with multiple mobile units, simultaneously.

This system, supported by GSM services, provides a mobility platform for low-bandwidth data. The DX protocol stack forms part of the middleware. This allows the top-level application (e.g. vehicle monitoring) to concentrate on the problem at hand. The protocol stack provides a reliable communication base on which these application services may be run.

This technology may be of use in several other mobile applications such as asset tracking and management (containers, etc.), wireless monitoring and control systems, personal applications, etc.

GSM SMS services and data services are accessed in different ways. There is no consistent access procedure to these services. The addition of the prototype stack has allowed the top-level fleet management to access services in a consistent fashion. Data services provide dedicated circuit-switch connection and limit data flow to the two parties. The SMSC functions in a store-and-forward fashion. SMSs are only forwarded to GSM mobile stations if they are currently registered on the network. SMSs are stored for a period until the client registers with the network. The GSM mobile stations are not consistent in this regard. Additional functionality must ensure storage of SMSs in the absence of GSM coverage. The introduced DX protocol stack relieves the burden on the upper layer application in this regard. The robust GSMFTP facility does high-level data flow control and error management and is ideal for true mobile applications. An interrupted GSMFTP data transfer procedure may be continued at a more convenient time.

The proposed DX protocol stack is implemented in a commercially available vehicle monitoring unit, of which thousands are in the field already, and a single central base management unit.

GPRS (**G**eneral **P**acket **R**adio **S**ervices) on GSM networks will remedy the data services circuit-switch approach and data services will provide higher bandwidth data channels. This will enhance flexibility when data is transported across a network.

Table 2 GSM Service Parameters

GSM Service	Latency	Rate	Destination	Applications
SMS	Few seconds to hours	Maximum of 140 octets every 2-3 seconds.	Determined at SMS transmission time	Very low data rate applications
Data Service	Average 100 ms	9600 bps	Determined at establishment of data switch circuit	Applications using data streaming services.

Chapter 4

WAP Protocol

The exponential growth of the Internet over recent years has turned it into an information library of global proportions with millions of people having access to it on a daily basis. Access was originally limited to users having fixed communication lines, but this is busy changing. The next big challenge is to provide the mobile community, of which mobile phone users forms a significant percentage, access to Internet services. The proposed **WAP** (**W**ireless **A**pplication **P**rotocol) stack aids wireless devices such as mobile phones, pagers, personal digital assistants, etc. to access the Internet.

The WAP Specification addresses mobile network characteristics and operator needs by adapting existing network technology to the special requirements of wireless data devices. Requirements of the WAP Architecture are to [49]

- Optimize for narrow-band, high latency bearers;
- Optimize for efficient device resource utilisation (low memory/CPU usage/power consumption);
- Provide support for secure applications and communications;

WAP promises to turn ordinary wireless devices into Internet-enabled devices. If wireless devices such as mobile phones may benefit from this specification, it is worthwhile to investigate. The objective is to search for functionality that can benefit operation and services provided by SUNSAT field stations. The ultimate aim is to turn the SUNSAT field station into an Internet-enabled device as well.

The WAP Protocol stack is illustrated in Figure 4-1.

Layer functionality will be discussed in the following subsections.

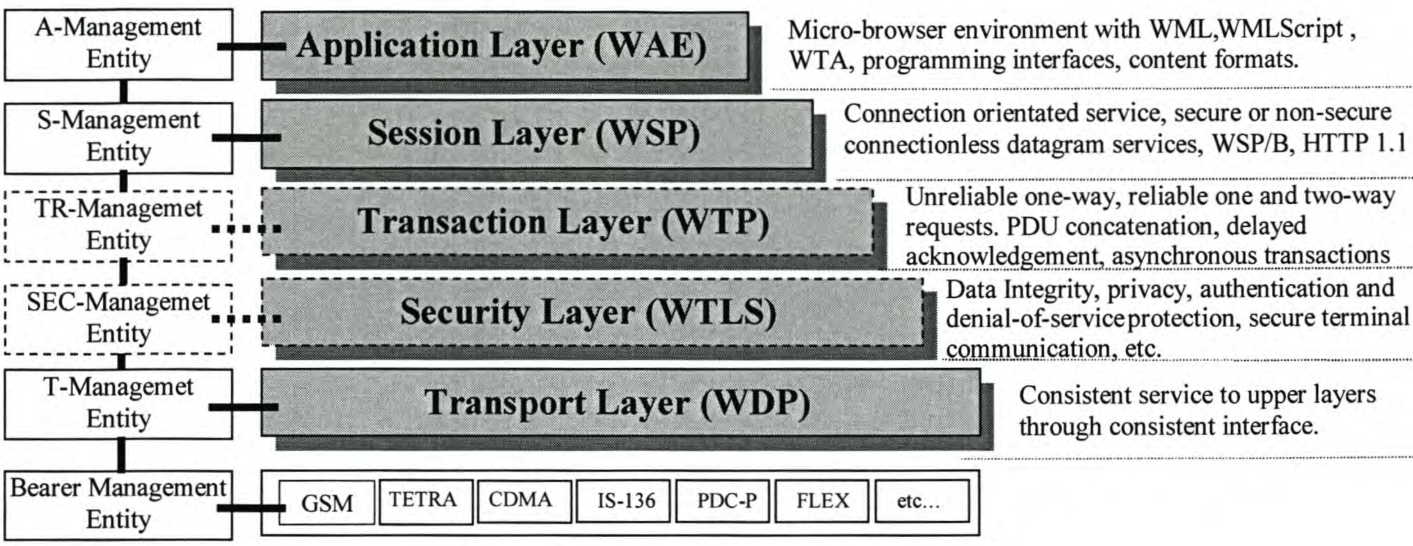

Figure 4-1 WAP Layers

The layered approach enables other services and applications to utilize features of WAP through a set of well-defined interfaces. External applications may access the session, transaction, security and transport layers directly. This allows the WAP Stack to be used for applications and services not currently specified for. Applications such as electronic mail, a calendar, phone book, notepad, and electronic commerce, or services, such as white and yellow pages, may be developed to use WAP. The protocols in the WAP family are designed for use over narrow-band bearers in wireless telecommunications

4.1 Wireless Datagram Protocol

The **WDP** (**W**ireless **D**atagram **P**rotocol) Layer operates above the data-capable bearer services supported by various network types. The Bearer Layer represents bearer services such as GSM SMS, or USSD, or IS-136 R-Data, or CDMA Packet Data. The bearer network is responsible for routing datagrams to the destination device. Addressing formats differ from one bearer to another (IP addresses or phone numbers). WDP is designed around an assumption that the operating environment is capable of transmitting and receiving data. A consistent service is provided to the WAP upper layer protocols (Security, Transaction and Session) which are designed to operate transparently over a variety of different bearer services.

WDP provides the upper layer protocols with services such as:

- application addressing by port numbers,
- optional segmentation,
- optional re-assembly,
- optional error detection.

Quality of service with respect to data throughput, error rate and delays may vary from one network bearer to another. Memory utilisation and transmission efficiency may vary between the different bearer implementations. WDP service and service primitives remain the same. This provides a consistent interface to the higher layers as previously mentioned. The different WDP services, their functionality and error processing are discussed in the following subsections.

4.1.1 WDP Functionality

WDP supports several simultaneous communication instances from a higher layer over a single underlying WDP bearer service. The port number uniquely identifies the higher layer entity above WDP. This may be another protocol layer such as the **WTP** (**W**ireless **T**ransaction **P**rotocol) or the **WSP** (**W**ireless **S**ession **P**rotocol) or an application such as e-mail. WDP routes an incoming datagram to the correct WDP user. This datagram service to the WDP user is provided regardless of the capability of the bearer network type. Most of the named bearer networks already provide such a datagram transport service. IP-based networks use UDP.

The Adaptation Layer is a sub-layer contained within WDP that maps the protocol functions directly onto a specific bearer. This adaptation layer is different for each bearer and deals with specific capabilities and characteristics of that bearer service. A complete set of specifications for each of these bearer services is published.

4.1.2 WDP Management Entity

The WDP management entity, the interface between the WDP layer and the device environment, provides information to the WDP layer relating changes in the device's environment which may impact on its correct operation. The WDP management entity will deal with all issues related to initialisation, dynamic re-configuration and resources, as they pertain to the WDP layer. The WDP management entity monitors the state of the following services:

- the mobile is within a coverage area applicable to the bearer service being invoked;
- the mobile having sufficient power and the power being on;
- sufficient resources (processing and memory) within the mobile for WDP operations;
- the WDP protocol correctly configures, and;
- the user is willing to receive/transmit data.

The WDP management entity notifies WDP if one or more of the assumed services are not available. If the mobile roamed out of coverage for a bearer service, the bearer management entity should report to the WDP management entity that transmission/reception over that bearer is no longer possible. The WDP management entity would then indicate to the WDP layer to close all active connections over that bearer. Low battery power would be handled in a similar way.

In addition to monitoring the state of the mobile environment, the WDP management entity may be used as the interface to the user for setting various configuration parameters used by WDP, such as device address. It may be used to implement functions available to the user such as a *drop all data connections* feature

The WDP management entity must interact with various components of manufacturer-specific communicating devices. The design and implementation of the WDP management entity is considered outside the scope of the WDP specification and is implementation-specific.

4.1.3 WDP Error Processing

WDP only generates errors when requested services are unavailable. These errors may be the cause of a wireless data gateway that cannot send a datagram to the WAP Gateway, no application listening to the destination port, or the receiver might not have enough buffer space to receive a large message.

The **WCMP** (**W**ireless **C**ontrol **M**essage **P**rotocol) provides an efficient error-handling mechanism for WDP resulting in improved performance for WAP protocols and applications. It is therefore advisable to implement WCMP.

4.1.4 WDP Conformance

A minimum set of WDP features must be implemented to ensure that implementations from multiple vendors will be able to operate with each other. Each of the bearer services, on which the WDP protocol operates, supports a datagram service. This datagram service is used to support the defined abstract primitives. For bearer services not supporting IP a specific WDP protocol is defined in [51] and MUST be used.

The following features are used within the WDP layer:

- Source Port Number is mandatory and must be included in send and receive operations;
- Destination Port Number is mandatory and must be included in send and receive operations;
- Segmentation and re-assembly on send and receive operations are optional;
- Text header information to be included on send and receive operations is optional;
- The mandatory *T-Unitdata* service primitive is used to requests services or used when issuing indications on requested services;
- *T-Derror* service primitive is optional when reporting errors on requested services.

4.1.5 WDP Bearer Dependent Profiles

The different bearer services are WDP over GSM (GSM SMS, GSM Circuit Switched Data), WDP over CDMA (CDMA Circuit-Switched Data, CDMA Packet Data) and WDP over TETRA (TETRA Short Data Service, TETRA Packet Data).

4.2 Wireless Transaction Protocol

The **WTP** (**W**ireless **T**ransaction **P**rotocol) requires datagram transport services and provides a light-weight transaction-oriented protocol that is suitable for implementation in "thin" clients or mobile stations. Throughout this text, the term *transaction* means a request by a client to a server including the server's reply to the original request. WTP protocol data is located in the data portion of the datagram. Since datagrams are unreliable, WTP is required to perform re-transmissions and send acknowledgements to provide reliable services to WTP users. WTP operates efficiently over secure or non-secure wireless datagram networks and provides the following features; [49]

- Three classes of transactions services:
  - Unreliable one-way requests
  - Reliable one-way requests, and
  - Reliable two-way request-reply transactions;
- Optional user-to-user reliability - WTP user triggers the confirmation of each received message;
- Optional out-of-band data on acknowledgements;
- PDU concatenation and delayed acknowledgement to reduce the number of messages sent; and
- Asynchronous transactions.

WTP provides services for interactive "***browsing***" (*request/response*) applications. During browsing sessions, the client requests information (WTP Initiator) from a server (WTP Responder), which MAY be fixed or mobile, and the server responds with the information

(Figure A.30). This *request-response* pair is referred to as a "*transaction*". Benefits of WTP include:

- Improved reliability over datagram services. WTP relieves the upper layer from re-transmissions and acknowledgements;
- Improved efficiency over connection-oriented service. WTP has no explicit connection set-up or teardown phases.
- WTP is message-oriented and designed for transaction-oriented services.

The different WTP transaction classes are briefly discussed in the following subsections.

4.2.1 Class 0 Transactions

Class 0 transactions provide an unreliable datagram service when an "unreliable push" service is required. Class 0 transaction behaviour is as follows: One *invoke* message is sent from the Initiator to the Responder as illustrated by Figure 4-2.

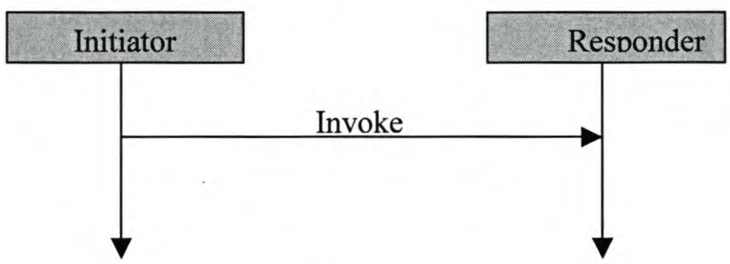

Figure 4-2 Basic Class 0 Transaction

The Responder does not acknowledge the invoke message and the Initiator does not perform re-transmissions. At the initiator the transaction ends when the *invoke* message has been sent. At the Responder, the transaction ends when the *invoke* has been received (see Figure A.31, Figure A.37).

4.2.2 Class 1 Transactions

Class 1 transactions provide a reliable datagram service. The basic class 1 transaction is as follows: One *invoke* message is sent from the Initiator to the Responder. The *invoke* message is acknowledged by the Responder. Figure 4-3 illustrates the two phases of a class 1 transaction.

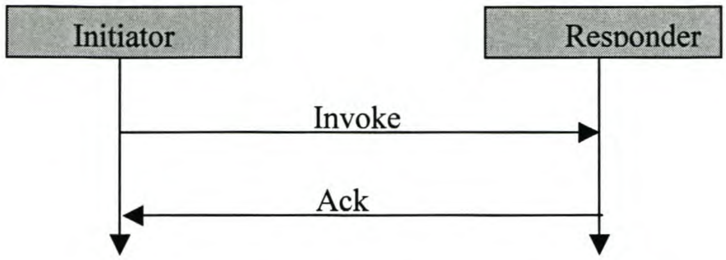

Figure 4-3 Basic Class 1 Transaction

The Responder maintains state information for some time after the acknowledgement has been sent to handle possible re-transmissions of the acknowledgement if it gets lost and/or the Initiator re-transmits the invoke message. At the Initiator, the transactions ends when the acknowledgement has been received. The transaction may be aborted at any time (see Figure A.31, Figure A.32, Figure A.37).

4.2.3 Class 2 Transactions

Class 2 transactions provide the basic *invoke/response* transaction service. Basic class 2 transaction behaviour is as follows: One *invoke* message is sent from the Initiator to the Responder. The Responder replies with exactly one result message that implicitly acknowledges the invoke message. This three phase Class 2 transaction is illustrated in Figure 4-4.

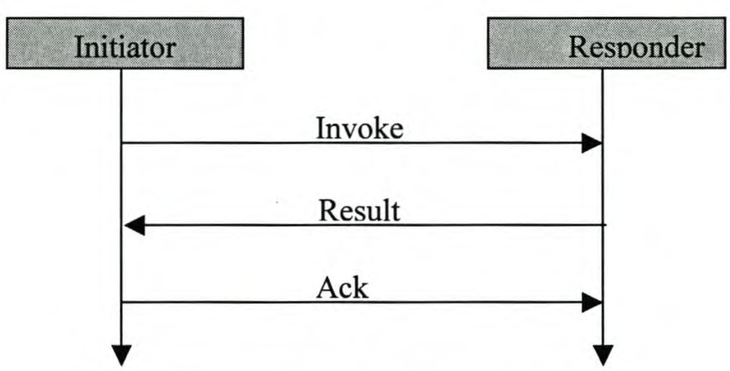

Figure 4-4 Basic Class 2 Transaction

If the Responder takes longer to service the *invoke* than the Responder's acknowledgement timer interval, the Responder may reply with *"HOLD ON"* acknowledgement before sending the result message. This prevents unnecessary retransmission of the *invoke* message by the Initiator. The four-phase Class 2 Hold-on transaction is shown in Figure 4-5.

The Responder sends the result message back to the Initiator. The *result* message is acknowledged by the Initiator. The Initiator maintains state information for some time after the acknowledgement has been sent. This is done in order to handle possible re-transmissions of the lost acknowledgement or the Responder re-transmits the result message. This transaction ends at the Responder when the acknowledgement has been received. The transaction can be aborted at any time.

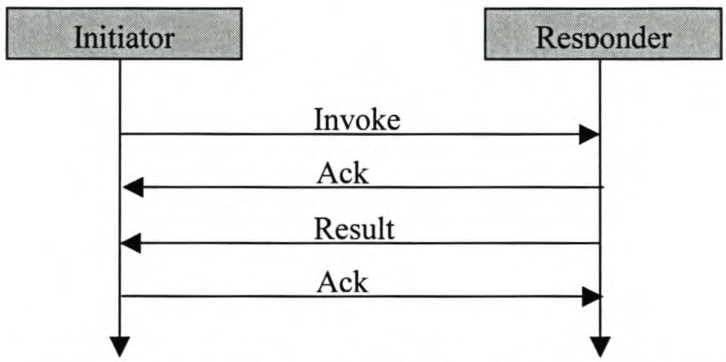

Figure 4-5 Class 2 Transaction with "Hold on" Acknowledgement

4.2.4 WTP Management Entity

The WTP management entity is used as an interface between the WTP layer and the environment of the device. The WTP management entity provides information to the WTP layer about changes in the device's environment, which MAY impact on the correct operation of WTP. If the mobile roams out of coverage for a bearer service, the bearer management entity SHOULD report to the WTP management entity that transmission/reception over that bearer is no longer possible.

4.3 Wireless Transport Layer Security

WTLS (**W**ireless **T**ransport **L**ayer **S**ecurity) is a security protocol based upon the industry-standard Transport Layer Security (TLS) Protocol, formerly known as **SSL** (**S**ecure **S**ockets **L**ayer). WTLS is intended for use with the WAP transport protocols and has been optimised for use over narrow-band communication channels. WTLS provides the following features [49]

- Data integrity - WTLS contains facilities to ensure that data sent between the terminal and an application server is unchanged and uncorrupted.

- Privacy - WTLS contains facilities to ensures that data transmitted between the terminal and an application server is private and cannot be understood by any intermediate parties that may have intercepted the data stream.

- Authentication - WTLS contains facilities to establish the authenticity of the terminal and application server.

- Denial-of-service protection - WTLS contains facilities for detecting and rejecting data that is replayed or not successfully verified. WTLS makes many typical denial-of-service attacks harder to accomplish and protects the upper protocol layers.

WTLS is an optional layer above the transport layer that preserves the transport service interfaces.

4.4 Wireless Session Protocol

The **WSP** (**W**ireless **S**ession **P**rotocol) is designed to function on transaction and datagram services (see Figure A.43). It provides the WAP Application layer with a consistent interface for two types of session services. The first is a connection-oriented service that operates above the transaction layer protocol WTP or WTLS. The second is a connectionless service that operates above a secure or non-secure datagram service (WDP) [49].

WSP currently consists of services suited for browsing applications (WSP/B). The protocols in the WSP family are optimized for low-bandwidth bearer networks with relatively long latency[49]. WSP does not require a security layer.

4.4.1 WSP Functionality

WSP provides the means for organised exchange of content between co-operating client/server application pairs. Applications can therefore:

(a) Establish and release a reliable session between client and server;
(b) Agree on a common level of protocol functionality using capability negotiation;
(c) Exchange content between client and server using compact encoding;
(d) Suspend and resume the session.

At the core of WSP is a compact binary form of **HTTP** (**H**yper **T**ext **T**ransfer **P**rotocol) that reduces protocol overhead. All defined HTTP/1.1 methods are supported. The capability negotiation is used to agree on a set of extended request methods so that full HTTP/1.1 compatibility may be retained.

WSP provides typed data transfer for the application layer. HTTP/1.1 content headers are used to define content type, character set encoding, languages, etc. with the better-known headers encoded in a compact binary format. A compact composite data format, that is semantically equivalent to MIME, provides content headers for each component within the composite data object. Sessions are not necessarily tied to the underlying transport. A session may be suspended to free network resources or save battery power. The re-establishment protocol resumes a suspended session without the overhead for normal session establishment. The session may even be resumed over a different bearer network.

4.4.2 Connection-Mode Sessions

The different connection-mode session service facilities are discussed in this section.

Session Management allows clients and servers to come to agreement on facilities and protocols to be used. Attribute information is exchanged during session establishment and remains valid throughout the session. Either the client or the server may terminate the session. The following service primitives support this facility [50] (see Figure A. 46):

- **S-Connect** primitive initiates session establishment and notifies of its success;
- **S-Disconnect** primitive is used to disconnect a session and to notify a session user that the session could not be established or has been disconnected.

Method Invocation permits the client to request the server to execute an operation and to return the result. Available operations are the HTTP (Get, Post and Reply [51]) methods or user-defined extension operations, which fit into the same request-reply. Service users, on the client or server, are notified about the completion of the transaction. The following service primitives support this facility:

- **S-MethodInvoke** requests server to execute an operation;
- **S-MethodResult** returns a the response to an operation request invoked by the S-MethodInvoke primitive;
- **S-MethodAbort** aborts an incomplete operation.

The service provider uses *Exception Reporting* facilities to notify the user about events not related to any particular transaction. The **S-Exception** service primitive is used to report these events.

Push permits the server to transmit unsolicited information to the client. This is a non-confirmed facility, so that delivery of the information may be unreliable. The **S-Push** service primitive supports this facility. **S-PushAbort** is used to reject a push operation.

Confirmed Push facility is similar to the Push facility, but the client confirms receipt of information. **S-ConfirmedPush** is used to send this unsolicited information from the server within the session context.

Session Resume Facilities allow both client and servers to suspend a session thereby preserving the state of the session. This mechanism is also used to cater for situations where further communication is no longer possible. It may be used to switch the session to an alternate bearer network. S-Suspend is used to request suspension of a session. S-Resume requests the resumption of a suspended session. Figure 4-6 illustrates the use of some of the S-Connect, S-MethodInvoke, S-Suspend, S-Resume and S-Disconnect primitives.

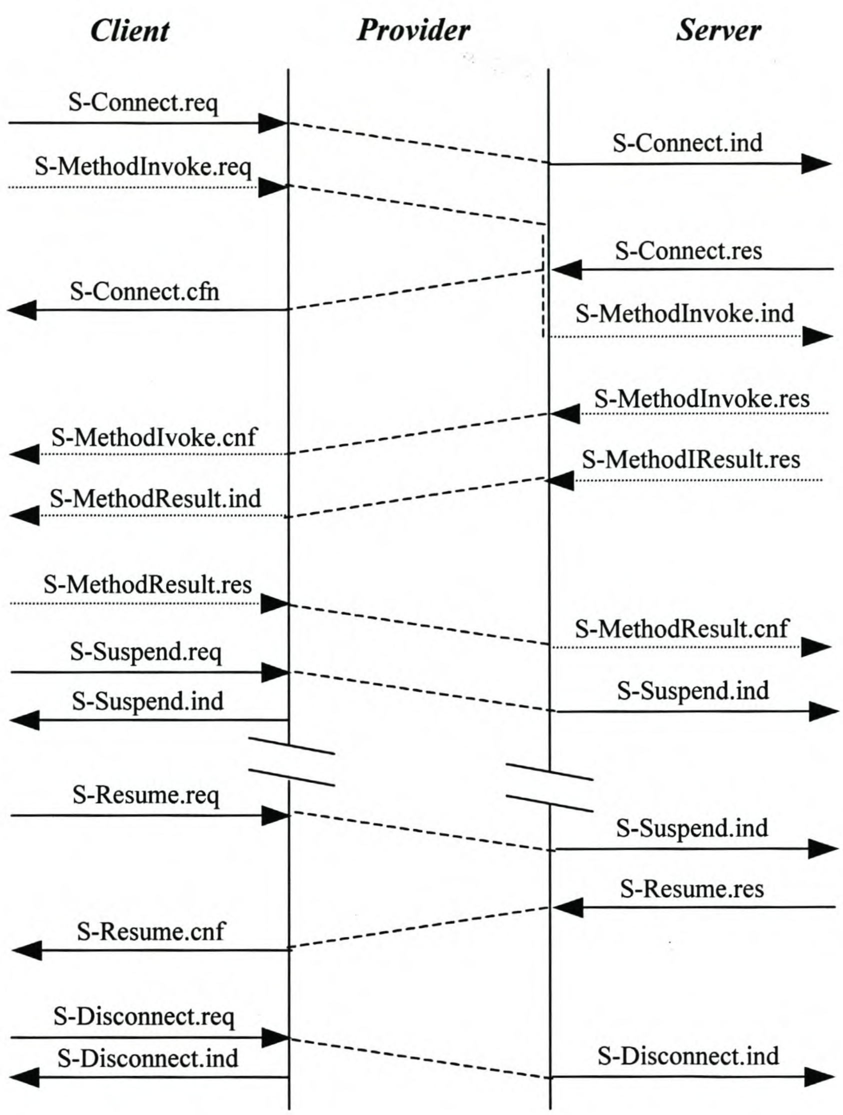

Figure 4-6 Service Primitives

4.4.3 Connectionless Sessions

Connectionless sessions use directly services provided by WDP or optionally WTLS. No state machines are used. Each primitive of the connectionless WSP service interface maps directly to sending a WSP PDU with the underlying *Unitdata* primitive.

4.5 Wireless Application Environment

The **WAE** (**W**ireless **A**pplication **E**nvironment) is a general-purpose environment based on a combination of **WWW** (**W**orld **W**ide **W**eb) and Mobile Telephony technologies. The primary objective of the WAE is to establish an interoperable environment that will allow operators and service providers to build applications and services that can reach a variety of different wireless platforms in an efficient and useful manner. WAE includes a micro-browser environment containing the following functionality:

- **WML** (**W**ireless **M**ark-up **L**anguage) – a lightweight mark-up language, similar to HTML, but optimised for use in hand-held mobile terminals. WML provides basic means for document formatting and user interaction but presupposes little of how they are actually implemented.

- WMLScript – a lightweight scripting language similar to JavaScript,

- **WTA** (**W**ireless **T**elephony **A**pplication) – telephony services and programming interfaces (WTAI); and

- Content Formats – a set of well-defined data formats, including images, phone book records and calendar information.

The WAP Programming model is similar to the WWW Programming model. Application developers benefit from the familiar programming model, a proven architecture and the ability to leverage existing tools. Optimisations and extensions match the characteristics of wireless environments.

The standard set of components include:

- Standard naming model: WWW-standard URLs identify WAP content and WWW-standard URIs identify local device resources.

- WAP Content typing is consistent with WWW typing. This allows for WAP user agents to correctly process content-based types.
- WAP Content Formats are based on WWW technology. They include display mark-up, calendar information, electronic business card objects, images, etc.
- Standard protocols communicate browser requests from the mobile terminal to the network web server.

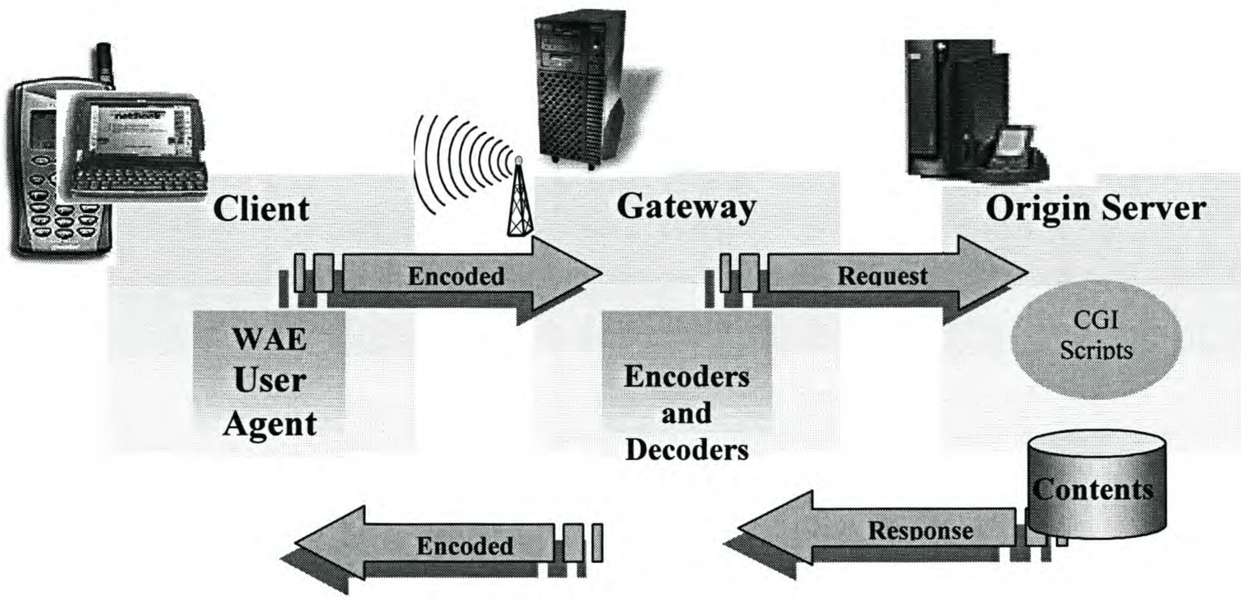

Figure 4-7 WAP Programming Model [49]

WAP Proxies connect the wireless domain with the WWW. WAP Proxies are comprised of:

- Protocol Gateways which translate WAP Protocol Stack requests (WSP,WTP, WTLS and WDP) to the WWW Protocol Stack (HTTP and TCP/IP);
- Content Encoders and Decoders that translate WAP content into compact encoded formats to reduce the amount of data transported over the network.

In Figure 4-8 the WAP Client communicates with two servers in the wireless network. In the Web environment separate specialised proxy servers may perform conversion, caching and content adaptation. The WAP proxy translates WML requests to HTML requests, thereby allowing the WAP client to submit requests to the web server. The proxy translates HTML to

WML and encodes these responses into a compact binary format to minimise information flow over low/medium wireless links.

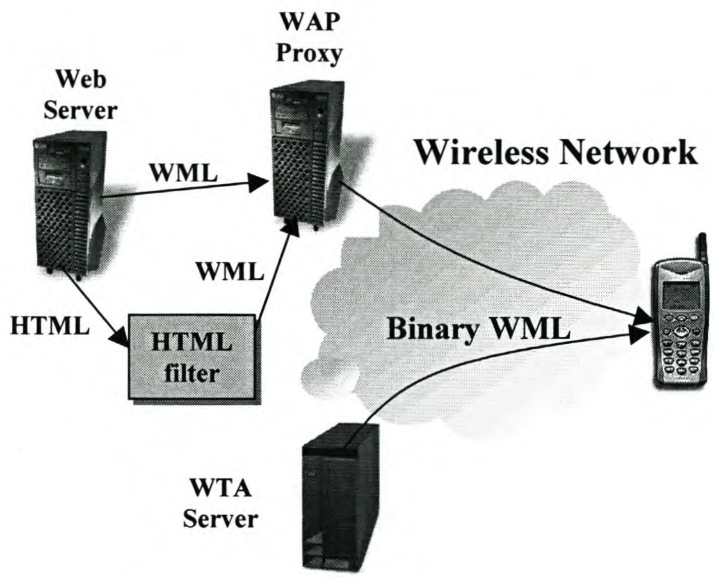

Figure 4-8 WAP Example

If the web server provides WAP content (e.g., WML), the WAP Proxy retrieves it directly from the web server. If the Web server provides WWW content (e.g., HTML), a filter translates the WWW Content into WAP content.

The WTA server is an origin or gateway server responding to requests directly from WAP clients. The WTA servers are used to provide WAP access to features of the wireless network provider's telecommunications infrastructure.

4.6 System Comparison

Having investigated WAP and having implemented two wireless communication systems, this section investigates their resemblances, differences and whether WAP may act as a practical middle-ware solution to their respective shortcomings. All three architectures are comparatively placed next to each other with reference to the OSI Model (see Figure 4.9)

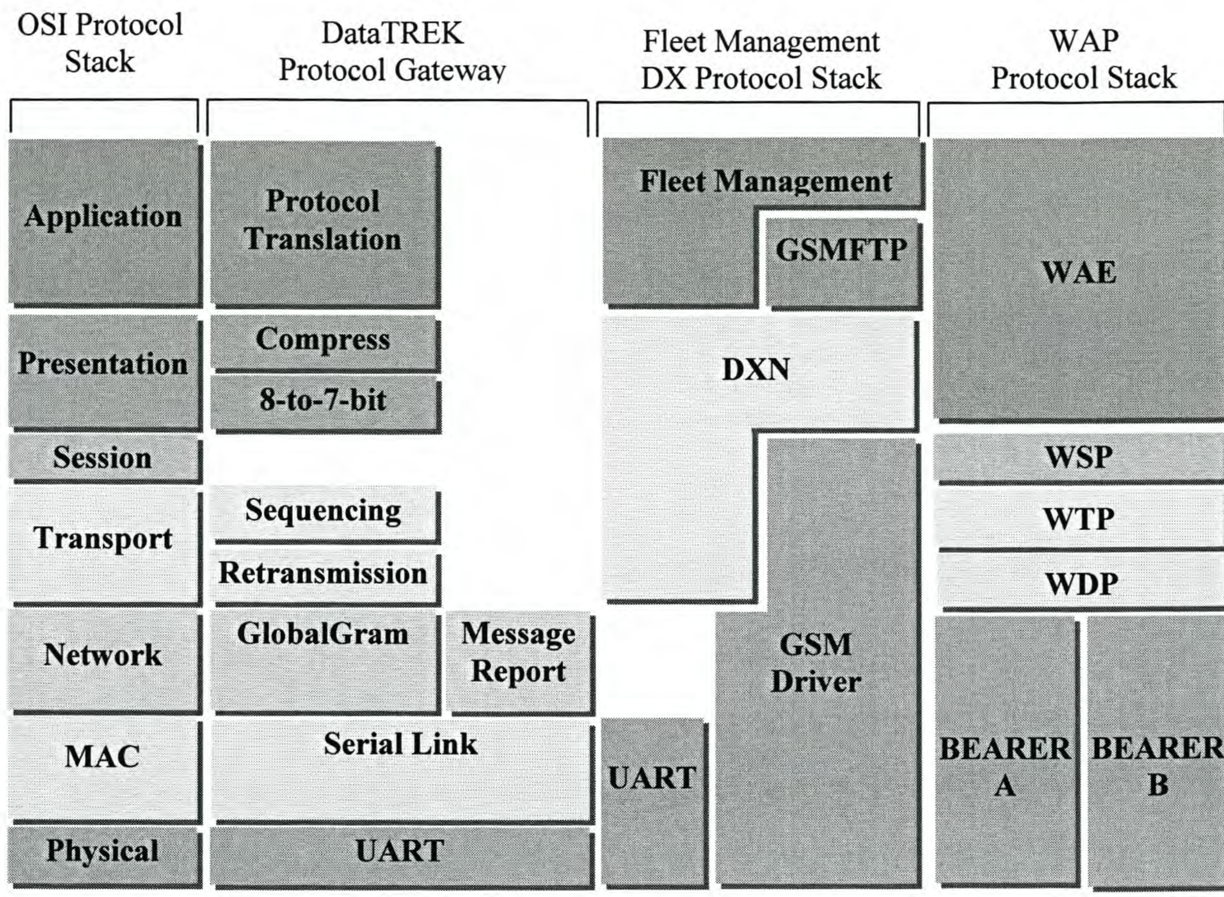

Figure 4-9 OSI Protocl Stack vs DataTREK Gateway vs DX Protocol Stack vs WAP

- There are different types of messages that may be sent to the actual communication device; namely; for
  - SCs there are GlobalGrams, data reports, commands and messages. GlobalGrams was the most suitable message type for the purposes of the DataTREK System.
  - The GSM Driver in the DX Protocol stack SMS may be sent in PDU mode, block mode and text mode. PDU mode was the most suitable and flexible for the purposes of fleet management.
  - The Bearer in the WAP product any datagram from SMS in GSM networks, UDP in IP networks, etc.
  - The WAP WTP Layer provides three different classes of transaction (0,1 and 2). Any of the transactions can be used depending on the importance of the data.

- The DX Protocol stack and WAP contains a multiplexing layer. The multiplexer layer allows layers higher in the protocol stack to operate on one or more different communication mediums
  - In the DX Protocol stack the GSMFTP and fleet management services operate in the same manner irrespective of whether they communicate using a dedicated serial port or via the GSM Network.
  - The WDP is an adaptation of the upper WAP layers to the bearer services. The WDP is designed to operate on several different bearers depending on what is implemented in the WDP layer.

- Robust services are provided by all three implementations:
  - In the DataTREK system lost GlobalGrams may be requested for retransmission.
  - GSMFTP on the DX Protocol stack will temporarily pause when a connection is lost and will continue from where it stopped once the data connection is re-established.
  - WSP may temporarily suspend a session when no bearer coverage is available only to be resumed later on when communication is possible again.

- Acknowledgement of successful completion of a transaction is issued in the DX protocol Stack and WAP.
  - In the DX Protocol stack the successful transportation of data may be viewed as a transaction. Indication to the fleet management software is given when such a procedure is completed successfully or unsuccessfully.
  - WAP transactions can provide different levels of acknowledgements depending on the functionality required.

4.7 Conclusion

The WDP layer introduced in WAP separates upper-layer functionality from the bearer network. This ultimately enhances re-use on different bearer services. Each layer's management entity is designed to be a network specific component and has to be re-evaluated for different bearers. Functionality of the management entity may therefore vary from one bearer service to another.

Since WAP is not necessarily bound to one bearer, it may even be implemented on the next generation of mobile networks. WAP is therefore not a transient or stepping-stone technology. As long as future networks have datagram facilities, WAP may be successfully implemented on these bearers. GPRS and UMTS, for example, may be targeted when this technology becomes available in different regions of the world.

Chapter 5

WAP/AX.25 and ORBCOMM

Commercial markets have earmarked WAP as a mobile-enabling technology primarily for mobile phones, PDAs, palmtop computers, etc. Although originally perceived as a technology for previously named devices, it can be implemented on different wireless communication devices. This chapter investigates the possibilities of implementing WAP on satellite mobile stations.

This chapter discusses the high-level concepts involved in implementing WAP in LEO store-and-forward systems. Limited WAP services or a sub-set thereof may be successfully integrated into LEO store-and-forward satellite communication systems. ORBCOMM may benefit from several WAP features such as end-to-end acknowledgement. The first section discusses what a WAP implementation on the ORBCOMM system entails. This is followed by an example on how WAP can be integrated into the SUNSAT operational environment.

5.1 WAP via ORBCOMM

This section discusses a solution to providing WAP services to ORBCOMM subscribers. These WAP services will not only allow ORBCOMM subscribers to have greater access to the Internet, but could have several other benefits. The WAP/ORBCOMM architecture is briefly looked at and the functionality of the additional components is discussed. This is followed by a short example and concluded by remarks on viability.

5.1.1 Benefits to Applications

Applications using the ORBCOMM store-and-forward facility use the SC's GlobalGram facility as explained in Chapter 2. Data is delivered as an E-mail message to its destination. Functionality to determine if messages have been successfully delivered or executed at the destination is not available. These additional features must therefore be designed into the application. If WAP middle-ware capable of acknowledging message delivery is added, the same application should be simpler in design.

Figure 5-1 ORBCOMM Network Inputs and Outputs

The DataTREK Protocol Gateway stores data for a fixed period before it is deleted. This allows the user to request retransmission of *lost* data. In a WAP-enhanced SC the DataTREK Protocol Gateway can transport the data via a WAP Class 1 Transaction. This unit, acting as initiator, keeps the transaction information until data delivery is acknowledged by the end-user, acting as the responder. As soon as the acknowledgement is received data is deleted from the Protocol Gateway's memory. When the user wants to remotely reconfigure the DataTREK Protocol Gateway, a Class 2 transaction might be used. The gateway, acting now as the responder, will acknowledge receipt of the request, and will set the requested features and finally reply with the result to the user, acting now as the initiator. These two simple scenarios illustrate how these middleware functions relieve the application from some overhead.

WAP port numbers allows different applications on the same device to be addressed individually in the same fashion as FTP, E-mail and TELNET applications on TCP/IP. On a normal SC, if such a feature is needed it would have to be designed into the application.

WTLS features such as security, authentication and privacy can automatically be used without the application being aware of it. The ultimate goal is to reduce the application complexity while making it more robust.

5.1.2 WAP/ORBCOMM Architecture

GlobalGram service is provided consistently throughout the world irrespective of a subscriber's location. GlobalGrams, as explained in Chapter 2, are self-contained data units that contain enough information to be propagated through the ORBCOMM System from its source to its destination.

Integration of WAP services with the ORBCOMM Network may be done in one of two ways. The first solution does not require any changes to the current ORBCOMM Network while the second solution does. The WAP/ORBCOMM architecture of the first solution is illustrated in Figure 5-2.

The ORBCOMM Network is used as transport medium for WDP **PDU**s (**P**rotocol **D**ata **U**nits) that are transmitted by the WAP mobile unit on the far right. This WAP mobile unit is able to communicate to the SC through the standard SC Protocol Interface. The data portion of the GlobalGram contains the WDP data unit. Information transported within the GlobalGram is not visible to the ORBCOMM System and cannot therefore be inspected or interpreted.

The device that is connected to the SC contains the WAP stack and the application. It is able to communicate to the SC in the standard GlobalGram fashion.

GlobalGram and messages are delivered via one of the satellites to the GES and GCC from whence it is forwarded to a remote third party E-mail server (see Figure 5-2). This WAP E-mail server processes ORBCOMM GlobalGrams arriving in the form of E-mail messages. WDP PDU information is extracted and further processed by the WAP E-mail Server.

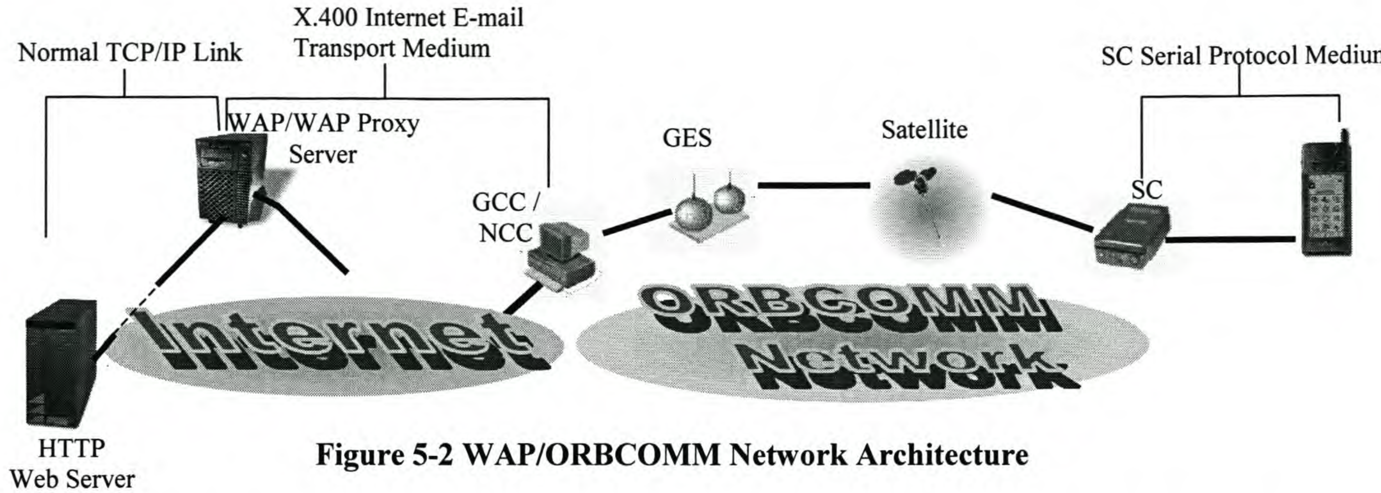

Figure 5-2 WAP/ORBCOMM Network Architecture

This E-mail server acts as a WAP server when it is the only server processing and responding to the WAP/SC mobile unit's requests. This E-mail server acts as a WAP Proxy server when another WAP or HTTP server on the Internet is accessed to execute the mobile unit's request.

5.1.2.1 WAP/ORBCOMM Proxy Server

The WAP/ORBCOMM proxy server acts as mediator between a WAP or HTTP server and the NCC. When communicating directly with a WAP server, the content-providing WAP server and WAP proxy exchange WML data. When communicating with an HTTP server, the WAP server translates requests and responses to/from HTTP1.1 and WML. Compact binary WML data are transmitted to the NCC.

This WAP/ORBCOMM Proxy server is a third-party service since it delivers a value-added service. This value-added service is therefore beyond the scope of ORBCOMM services.

5.1.2.2 ORBCOMM SC Addressing

There are several addressing mechanisms used within the ORBCOMM Network. The two of interest are:

- ORBCOMM speed dials or indicators, and
- E-mail addresses.

Speed dials or indicators are numbers that are associated with a specific E-mail address. This association is specified when an SC is registered on the ORBCOMM Network and kept on record at the NCC. Speed dial indicators occupy only one byte in a GlobalGram which results in a dramatic saving compared to the long E-mail address format. The use of speed dials limits the subscriber to only a few predetermined destinations. Complete E-mail addresses allow for greater flexibility since the destination is determined at transmission time. The result is greater inefficiency since the overhead (ratio between user data transmitted and total data transmitted) is smaller. The ORBCOMM speed dial is a valuable feature that not only result in a saving on overhead but is simpler to implement.

5.1.2.3 WAP/GlobalGram Example

A WDP PDU is tunnelled inside a GlobalGram as illustrated in Figure 5-3. The GlobalGram PDU structure is shown at the far left. A normal GlobalGram uses speed-dial address indicators and not complete E-mail addresses. This results in a saving of several bytes.

There is no official WDP specification for ORBCOMM GlobalGram Network Services. The specification designed for use in SMSs can be used.

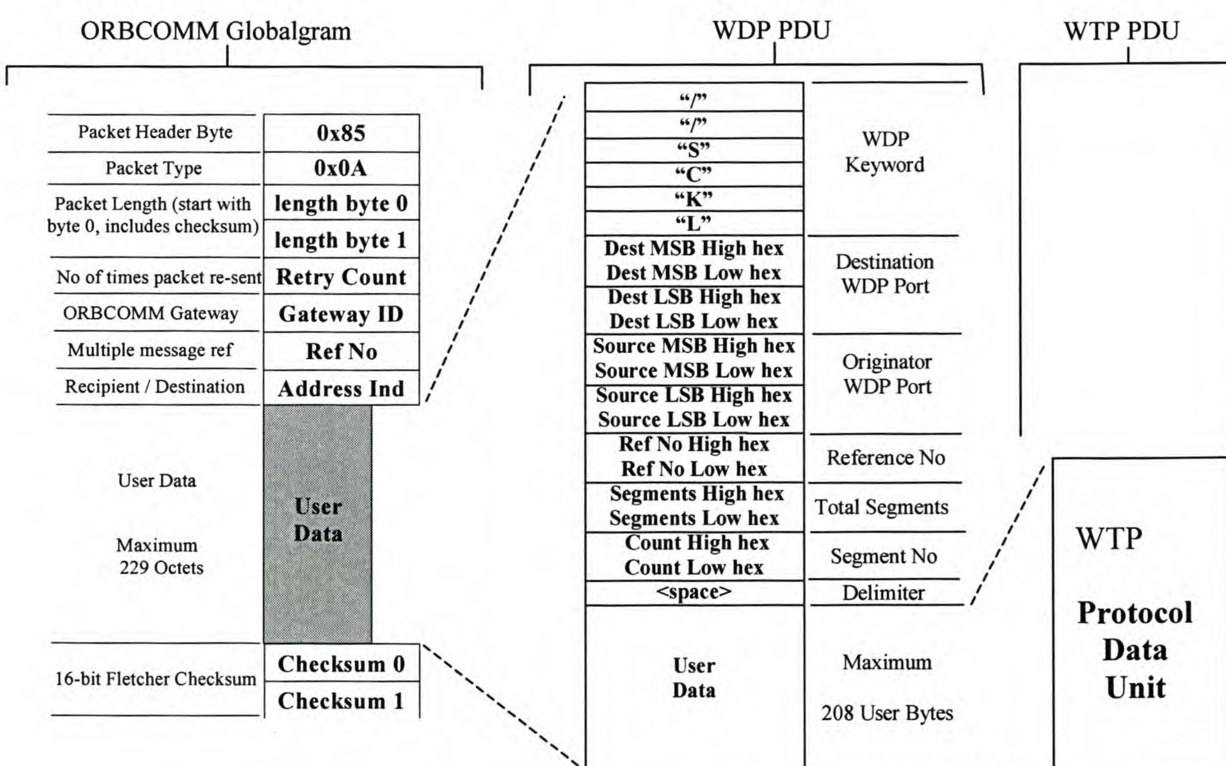

Figure 5-3 Globalgram transporting WDP PDU

The WDP has a fixed prefix "//SCKL". The WDP SMS PDU specification accepts only alpha-numeric as valid data. Control information and user data are therefore divided into nibbles. Each nibble is then stored as a hex digit '0'-'F'. This encoding and decoding follows the same principles when preparing SMSs for transmission as explained in Chapter 2. This is not the most efficient representation of data but it is the most flexible. The DataTREK data collecting system, the e-mail server and the SCs are configured to process only a limited character set (7-bit data). This is a more flexible option since most systems will accept this data form without any problem. These systems can be configured to accept binary data but this is not always possible since it is not under direct control of the user. Configuration to accept binary data is more cost-efficient but may limit data distribution to systems configured for 8-bit data.

The WDP structure contains the following control information: the destination port, the originator port, reference number, total segments and segment number. The destination address is contained in the GlobalGram control information. The E-mail message delivered to the end-user contains the SC identification from whence the message originates in the subject portion of the message. GlobalGrams may contain a maximum of 229 user bytes. The WDP header consists of a maximum of twenty-one bytes, which leaves 208 WDP characters or 104 octets to be transmitted by the WDP.

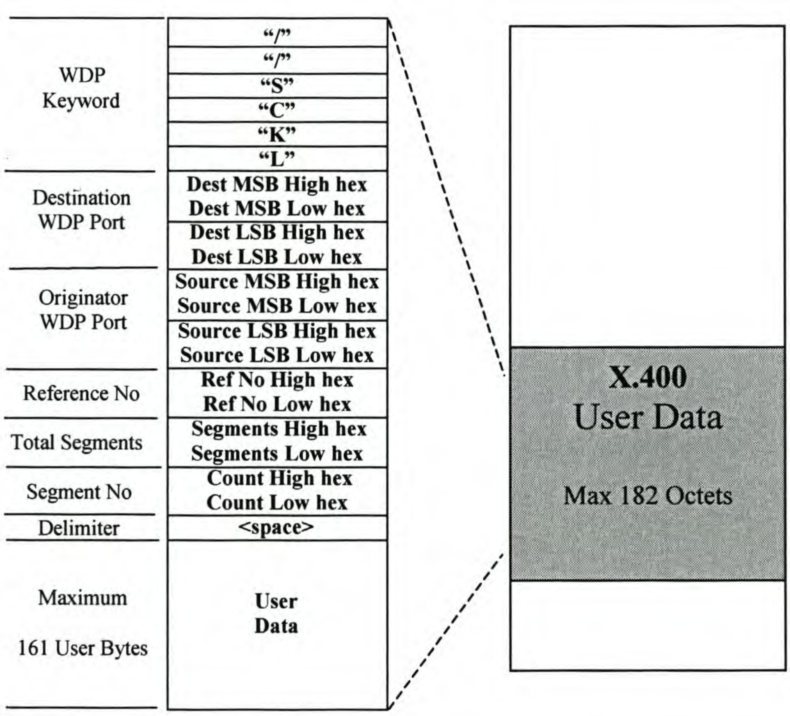

Figure 5-4 WDP Structure of E-mail messages

The ORBCOMM network is asymmetric when looking at the maximum number of user bytes that can be transmitted. A GlobalGram, transmitted from the SC to the E-mail server, may contain a maximum of 229 user bytes, whereas E-mail messages transmitted from the E-mail server to the SC, may contain a maximum of 182 user bytes. With 21 WDP header bytes, only 161 alphanumeric bytes can be contained in the WDP.

5.1.2.4 Disadvantages

In the architecture discussed thus far (illustrated in Figure 5-2) a third party provides specific subscribers with WAP/WWW services via a WAP Proxy or WAP server. This is not the ideal solution, because:

- WAP Proxy servers are introduced that are physically separated from the GCC/NCC which adds additional delays and leads to a longer communication path.
- The Internet is used to carry WAP content in e-mail messages to the third party WAP Proxy server, which is a slower transport medium.
- GlobalGram services in most cases introduce a considerable delay where mobile units are not in close proximity to a gateway.

5.1.3 Alternative Architectures

An alternative architecture exists which totally removes the need for a third-party service provider which operates independently from ORBCOMM. This architecture, illustrated in Figure 5-5, assumes that these services are directly provided by ORBCOMM. .

The result is a shorter communication path and removes the extra delay introduced when data is forwarded via the Internet from the GCC/NCC to a third party WAP Proxy. ORBCOMM has direct control of the WAP Proxy and may therefore configure it to integrate more effectively.

Additional functionality may be added to the SC that will accept WAP (WDP, WTP, WSP) requests. Bandwidth is therefore utilised more efficiently. Both ORBCOMM and SC participation is required for successful implementation.

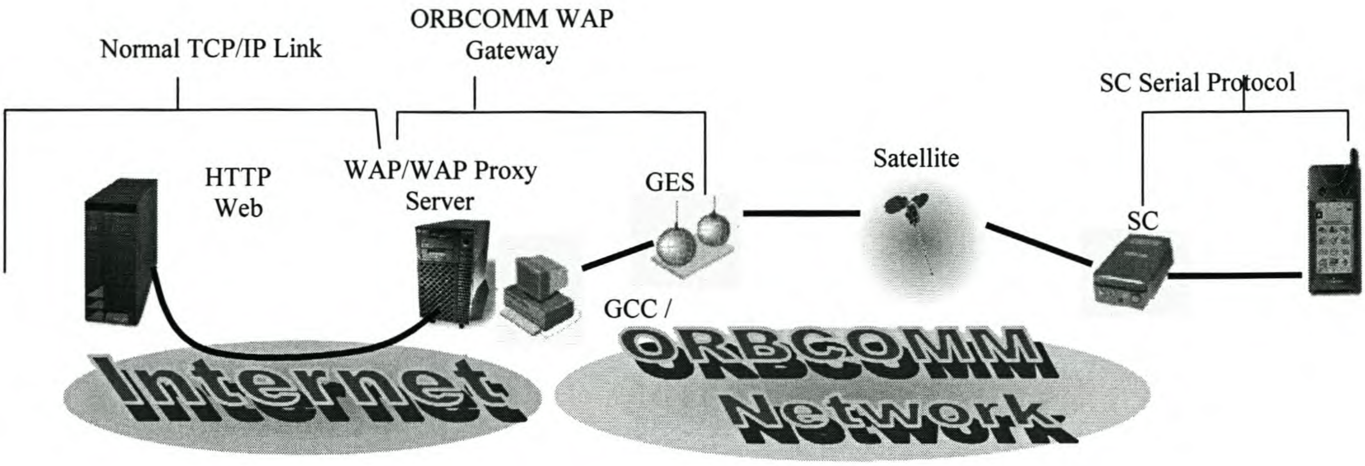

Figure 5-5 Alternative WAP/ORBCOMM Architecture

5.1.3.1 Advantages

- These suggestions result in more efficient use of bandwidth between the mobile unit and the SC since no WAP WDP PDU need to be transmitted on the serial link anymore.

- The communication path and the delay between the mobile unit and WAP Proxy server is decreased when this server is directly integrated into the ORBCOMM Network,

- A one-stop service may be provided by ORBCOMM to its customers. No third-party service provider is involved.

5.1.3.2 Viability

The ORBCOMM network is not recognised as a bearer service in the WDP specification. By including the ORBCOMM network as a bearer service, the WDP structure may be streamlined to contain control data specific to the ORBCOMM network. This will result in a further saving of bandwidth.

Delay, between submission of a message to an SC and the delivery to end-users, in certain regions of the world may be a number of hours. These circumstances do not allow delivery of near real-time interactive WAP services. ORBCOMM subscribers who are in close proximity to a GES may have access to near real-time interaction because of the additional available non-store-and-forward services. Hence delay may be reduced from hours to seconds.

5.1.4 Conclusion

WAP technology discussed in this subsection appears to be a promising opportunity to ORBCOMM subscribers for increase Internet access. WAP is designed to efficiently use low-bandwidth potentially high-latency wireless networks. The ORBCOMM network meets these criteria.

The third party WAP service provider solution is not the ideal solution. ORBCOMM and the SC manufactures need to get involved to make it a viable technology. SC equipment manufacturers, such as Stellar, have facilities to provide software upgrades via the Internet to existing SC equipment. SCs already in use can therefore be turned into WAP enhanced SCs at a later stage.

5.2 WAP via AX.25

LEO satellites have provided global store-and-forward communication services since the 1980's. This technology enables users to relay data between virtually any two locations on earth. The AX.25 protocol allows users to communicate with the satellite using standard packet communication. The PACSAT Protocol Suite uses these AX.25 services, thereby transforming these satellites into orbiting file servers. Users could for the first time directly interact with satellites by paging through the on-board file server's directories and uploading/ downloading digital files.

This introduced a slight shift in functionality from traditional communication satellite operations: it is now not only a direct relay switch between two or more earth-bound locations, but also a digital file store as well. As advances are made, more operations, traditionally executed by earth-bound servers, will be introduced into on-board satellite operations.

WAP promises to deliver limited Internet services over high-latency, bandwidth-limited wireless networks. Although specifically earmarked for use on mobile phones, it lends itself to a range of different mobile devices.

Broadband LEO communication satellite constellations promise to deliver Internet services to mobile users. Narrow-band micro-satellites can support limited services. The introduction of WAP on a satellite may expand its services. Files may now be accessed in a similar fashion as on the Internet. Authorisation, authentication and other standard security features combined with transaction services may deliver electronic commerce (e-commerce) to previously excluded users.

5.2.1 Applications

A considerable delivery delay of up to several hours is introduced when store-and-forward services are used on a single LEO satellite. Applications such as metering systems, semiautonomous and autonomous control systems that can tolerate such a delay can certainly benefit from such services. The ORBCOMM Network has successfully provided global two-way communication services such as tracking, global e-mail, assets monitoring and global messaging to their subscribers for the past number of years [33].

5.2.2 PACSATs

PACSAT is a generic term in the amateur radio service for a low earth-orbiting spacecraft which carries a large on-board memory for the purpose of data storage and retrieval by ground stations [37]. A PACSAT is a multi-channelled, full duplex device, with short periodic access times dictated by orbital mechanics. The PACSAT Protocol Suite, a product of the University of Surrey [37][54], was developed during the early 1990's. Its main aim is to efficiently share limited bandwidth among users that are much more than the number of available uplinks and downlinks. The PACSAT Protocol Suite uses AX.25 services to broadcast the file directories [37] and to provide file transfers[54]. Users could, for the first time, directly interact with satellites by paging through the on-board file server's directories and the uploading/ downloading of digital files such as telemetry, digitised voice, video images, personal mail and forwarded mail. Some of the features of PACSATs are [37]:

- Forwarded mail messages. These are messages not destined for PACSAT as an endpoint, but which are in transit between forwarding gateway stations.

- Personal electronic mail messages. These are messages to and from individuals who are using the satellite as a **BBS** (**B**ulletin **B**oard **S**erver). These messages use PACSAT as an endpoint. It can be argued that all mail should be forwarded mail, that no one use PACSAT as a direct BBS. There are three counter arguments:

    - For access in remote areas without a terrestrial infrastructure in place.
    - For emergency access by minimal ground stations.
    - To permit large numbers of people to have a direct hands-on experience with satellite communications

- Real-time Telemetry. These are current spacecraft telemetry values, such as solar array power, internal temperature, etc.

- Stored Telemetry. This is a file of one or more telemetry values stored over time, for example, the output of the solar arrays once a second over the last orbit. This is usually called "Whole Orbit Data" or "WOD".
- Bulletins. These are items of general interest, orbit predictions, AMSAT, News, Gateways, etc.

- Point-to-multipoint messages. These are messages which one PACSAT user wishes to send to several other PACSAT end users. This may be implemented using "mail groups" or "CC" lists

The PACSAT protocol suite consists of two protocols, namely the PACSAT Broadcast Protocol (PBP) and the File Transfer Level 0 (FTL0) Protocol. These protocols reside in the information field of the AX.25 Packet-radio link-layer protocol. Currently the protocol is only used for uploading of files to the satellite server and the PBP protocol is used for broadcasting the directory and files. The uploading of files is a point-to-point transaction, but any information being sent from the server is broadcasted.

5.2.2 WAPSATs

PACSATs provide users with a directory of on-board files from which the user can select files for download. WAP-enabled devices communicating with a WAP-enabled satellite, called WAPSATs, have similar capabilities as a PACSAT. WML may therefore be used which is a flexible solution to providing different types of data to the field station.

WAP operates on bearers that have datagram delivery capabilities. Unreliable datagram delivery is more than sufficient. Bearer session services are not a prerequisite.

5.2.2.1 Amateur X.25 Protocol

The AX.25 protocol allows satellite users and operators to digitally communicate with amateur satellites. AX.25 has two operational modes; namely:

1. **Connection-orientated** AX.25 sessions reliably relay data between the satellite and its users. Data delivered via these sessions is delivered without error and in sequence. Sessions are the preferred medium of communication.

2. The **connectionless mode** is at best an unreliable service. Data may be relayed between the satellite and its clients without establishing AX.25 sessions, but successful delivery of data is not guaranteed.

Since AX.25 has datagram delivery capabilities, it may even act as a WAP bearer service. AX.25 may then be dramatically scaled down since the connection-orientated services are not required by WAP. WSP manages WAP sessions. AX.25 **UI** (Unnumbered **I**nformation) frames delivers data in an unreliable fashion between the satellite and one or more clients. UI Frames may therefore be used to deliver WDP information between the satellite and its clients. The following illustration, Figure 5-6, shows the structure of UI frames and how they convey WAP information. The WTLS layer is not included in Figure 5-6 in order to keep the illustration simple.

The AX.25 frame contains a PID field that determines the type of data that is contained by the information field. This field distinguishes between PACSAT (PBP and FTL0), TCP/IP, AppleTalk ARP, etc. WAP may be added to this list of identifiers. This feature allows one single AX.25 protocol stack to act as bearer service to both the PACSAT Protocol and the WAP Protocol Stack.

The **SSID** (**S**econdary **S**tation **ID**entifier) in an AX.25 packet header may be used to distinguish between up to sixteen different applications on AX.25. The PBP server and FTL0 server are stations 11 and 12 respectively.

Multiple WTP PDUs may be contained in a single WDP PDU. This adds to efficiency by minimising the number of packets. The impact of the overhead involved in AX.25 frames (destination and source addresses, checksums, etc) is therefore minimised as well.

WML is based on HTML and is WAP's equivalent to XML. XML is meta-programming language that describes the data that accompanies it. This is a universal method of exchanging data since data is accompanied by a description of its structure. A WML source file, which is a text file containing the data structure description, must be parsed and compiled into a WML binary format. This WML binary is a shorthand, compressed equivalent to the WML source file. This WML binary is then inserted into WSP/WTP frames that results in a further saving in bandwidth.

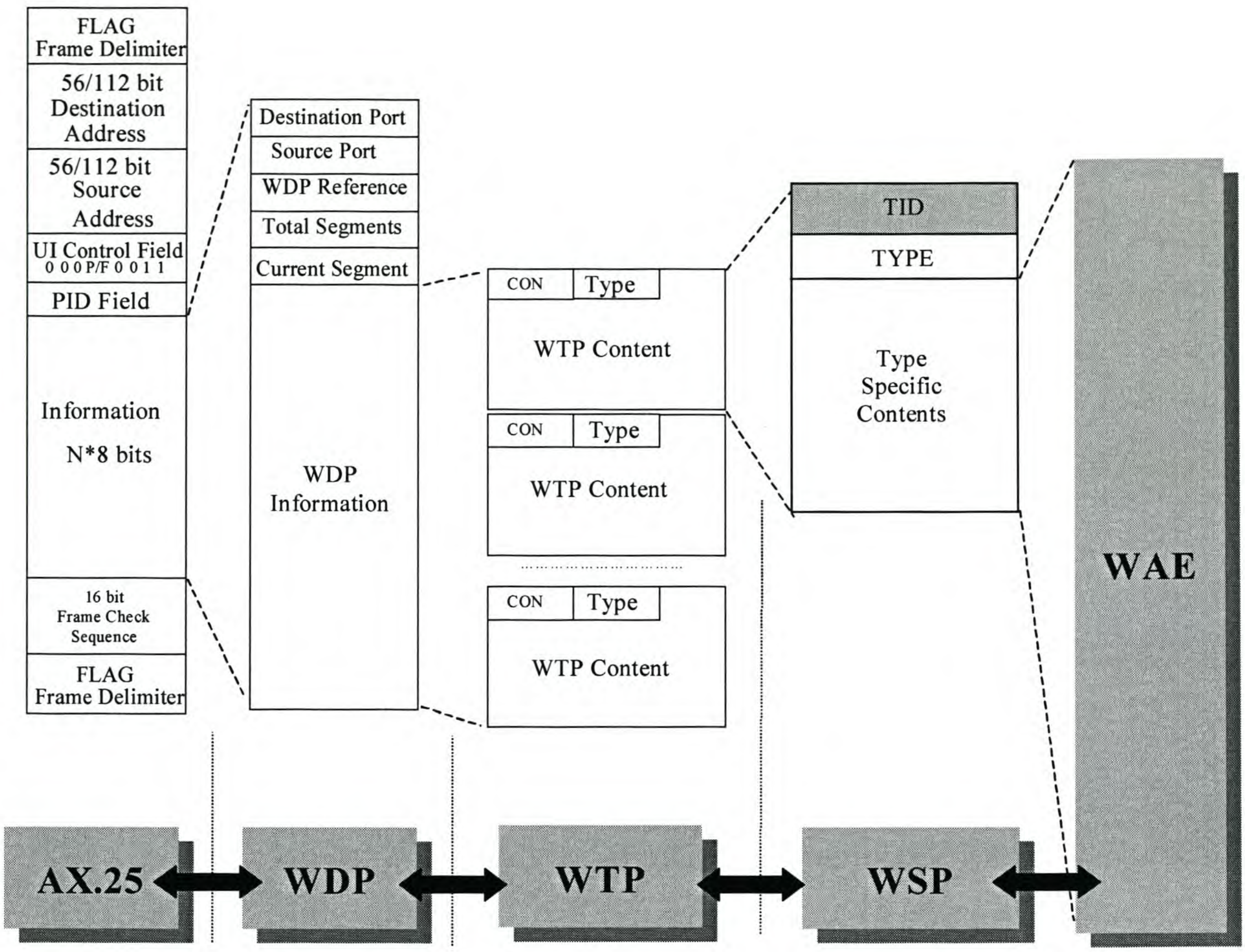

Figure 5-6 WAP information tunnelled in AX.25 UI Frames

5.2.3 Benefit of WAPSATs

The standard naming model is used. WWW-standard **URL**s (**U**niversal **R**esource **L**ocators) identify WAP content and WWW-standard **URI**s (**U**niversal **R**esources **I**dentifiers) identify local device resources.

This allows users, without radio communicating equipment but with links to the Internet to download/upload files from/to WAPSATs via a satellite WAP Proxy. Global file distribution is therefore not limited to individuals with radio equipment, but to every individual who has access to the Internet.

WTLS features include data integrity, privacy, authentication, denial-of-service protection support and secure communication between terminals. Data can now be exchanged between two or more users in a secure fashion.

WTP provides support for **RPC** (**R**emote **P**rocedure **C**all) and method invocation. With satellite-to-satellite communication, a satellite may request a neighbouring satellite to execute a certain procedure with the parameters provided and to return the results. The scenario may even be taken a step further: a satellite may request the control ground station or other terrestrial-bound devices to execute certain procedures under specific circumstances. This typically may happen when a specific satellite does not have the resources available, whether it is processing power or otherwise, to execute the requested procedure.

For data to be reliably delivered to a PACSAT, a connection must first be established. WTP supports reliable and unreliable delivery of data between a WAP server and its clients in a connectionless fashion. Connection set-up and break-up procedures are bandwidth-and time-consuming. This is especially true when large number of users require reliable data delivery of only a few bytes. In these circumstances, connection set-up and break-up may even consume more bandwidth than the data itself. WTP can therefore support more users requesting reliable data delivery of only a few bytes. This therefore forms the platform for e-commerce or transaction-orientated applications.

When users share the satellite communication footprint with the satellite WAP proxy server, the satellite may act as a relay between the proxy server and mobile WAP units. The proxy server may even act as a gateway to the rest of the Internet. If the web server provides WAP content (e.g., WML), the WAP Proxy retrieves it directly from the web server. If the Web server provides WWW content (e.g., HTML), a filter is used to translate the WWW content into WAP content.

If PACSAT and WAP share the same on-board file servers, Internet users may place a file on the satellite, that may be downloaded by a PACSAT user at a later stage. This WAP platform is a middle-tier development platform that should allow the delivery of functionality to mobile devices more quickly.

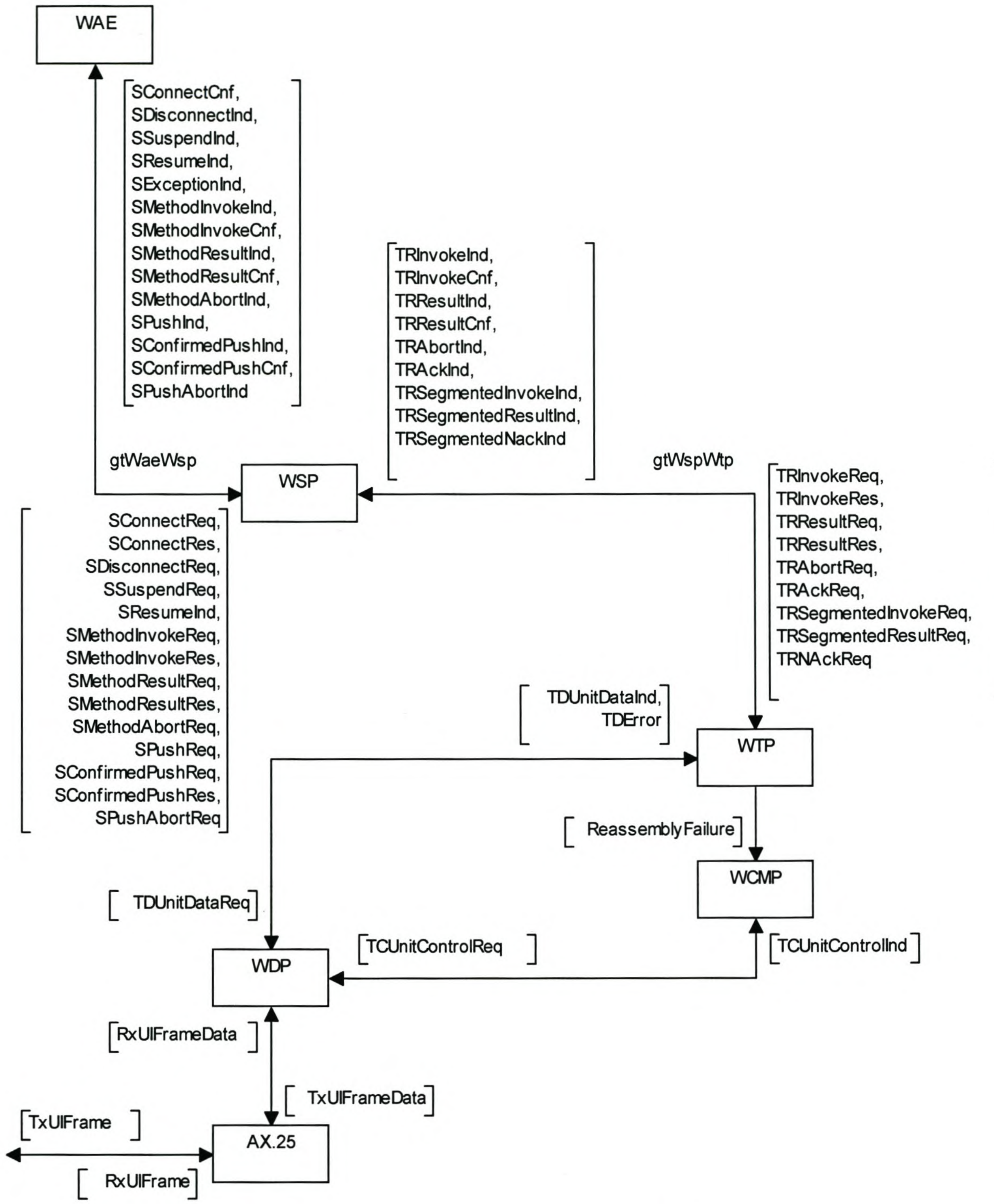

Figure 5-7 SDL System Diagram of WAP/AX.25

5.3 Conclusion

WAP is an open architecture that has the potential to turn wireless communication devices into Internet-enabled ones. Satellite wireless communication devices may benefit from WAP features as well. This chapter showed that WAP implementation in a store-and-forward LEO satellite environment is within reach. It is a viable solution to add the rich set of WAP features to mobile stations. Different solutions were given towards implementing WAP on the ORBCOMM Network and within the SUNSAT environment.

WAP is designed around a request-respond transaction scenario. The Internet WEB services operate in this manner. Several new features such as remote procedure calls, high volume of small data transactions and addressing mechanisms used within the Internet community can change the store-and-forward server into a WAPSAT. A WAPSAT is a store-and-forward satellite with a WAP protocol communication stack that is capable of processing transactions as requested by field stations. Since this chapter has looked at the overall satellite network architecture, the next chapter looks at the SUNSAT field station and how it can benefit from WAP middleware.

Chapter 6

SUNSAT Mobile Stations

A single LEO satellite, like SUNSAT, allows mobile field stations to transport data in a store-and-forward fashion between virtually any two locations on earth. Although global communication is possible, the end-to-end delivery time may vary from several seconds to several hours. Bandwidth is a scarce resource that must be optimally employed and for this reason an efficient protocol is required for narrow-band services. Multiple users may contend for a single data channel and an appropriate multi-access protocol is therefore required. The physical SUNSAT radio equipment and SRMA MAC protocol were therefore investigated. Data link layer functionality is not sufficient to allow for flexibility and reliable data delivery. Additional functionality can enhance the communications protocol stack tremendously. Applications may benefit from services such as compression, sequencing, store-and-forward buffers, different message types, etc. as seen in previous chapters.

Middleware must be able to provide a range of services to different applications over these narrow-band channels. The proposed functionality is discussed in this chapter. The satellite must be able to support high transaction volumes. These transactions exchange small amounts of data. Some communication networks favour long, continuous data transmissions whereas others favour short bursts of information. Possible applications are first discussed, followed by the description of the proposed solution that can support these applications.

The DataTREK, the Fleet management and WAP implementations are compared and evaluated. The comparison and recommendations are then used to determine if it is useful for implementation in SUNSAT field stations.

6.1 SRMA MAC Layer

SRMA uses a dynamic reservation system when allocating upload time to individual mobile stations that contain data in their transmission queues [27]. Mobile stations request transmission time from the satellite whenever they have a message packet to transmit. The satellite schedules the mobile station for communication when it receives the request. The satellite maintains a queue of requests and informs the users when they are scheduled.

To prevent collisions between request and message packets, the channel is either time divided or frequency divided between the two types of data. Time division multiplexing is a more cost-effective and simple solution than frequency division multiplexing.

The request channel can function in a fixed assignment or random access mode. Fixed assignment places an upper limit on the number of users in the system. For this reason the request channel is divided into fixed-size slots in a manner similar to slotted ALOHA. SRMA uses a fixed assignment method to achieve maximum throughput of requests.

During communication, the order of events is as follows:

- The protocol cycle begins when the satellite transmits a polling signal. The signal precedes the time slots in the request channel and notifies the users of the satellite's presence. Signalled users with messages to transmit randomly select one of the time slots and transmit a request. Requests transmitted in the same time slot will collide and are subsequently ignored by the satellite. A queue of all the successful requests is formed after the last time slot has expired.

- Users are notified one at a time to transmit their messages until all the users in the queue have been serviced. This marks the end of a cycle and the satellite transmits the polling signal to initiate the next cycle. Users who had collisions can now re-transmit their requests.

The ORBCOMM Air Interface Specification (see example Appendix D) operates on the same principles contained in the SRMA MAC Protocol Layer. SRMA is more robust for use in low data rate channels [27] because of increased transmission delays at high baud rates. The delay decreases with a decrease in data [27]. A compromise must be made between low delay and high effective data throughput.

SRMA and ORBCOMM's Air Interface operate on the same principles. The effective throughput and delay would therefore be comparable. SRMA as designed for the SUNSAT field station, only uses time-division multiplexing. The ORBCOMM Air Interface uses both a TDMA and FDMA access scheme. ORBCOMM Satellites have six receivers dedicated to SC subscriber communication. The ORBCOMM System will have better throughput than SUNSAT's SRMA because of greater access to resources such as bandwdith. ORBCOMM's ability to identify a channel with the lowest noise levels further enhances communication. SRMA, is therefore a simplified version of the ORBCOMM Air Interface Protocol.

Data link functionality alone is not very flexible and cannot provide the following services:

- The satellite assigns a communication slot to a mobile station. Only one logical file can be created containing all of the data transmitted during the assigned communication slot. Different types of data to different destinations can not be transferred during one session.
- Acknowledgement of individual portions of data.
- Simultaneous transmission of multiple streams of data.

6.2 Field Station Applications

Different control and communication systems demand different levels of feedback. The different reaction times of different systems ultimately determine the type of network they may be connected to. A billing system, for example, needs a continuous link until the billing transaction is completed. Credit card authorisation systems must have access to relevant data in order for a transaction to be authorised. Another class of systems, namely metering, E-mail and data gathering systems, do not need continuous access to network services. Access to data communication networks is needed only sporadically or periodically with less stringent reaction times. Electricity meter readings are only taken periodically for example once a month. Remote, autonomous control systems can be included into this category of applications as well since they only communicate periodically and need low response times.

The required bandwidth determines a suitable transport medium. Systems with a sporadic or periodic information generation pattern, in most cases are narrow-band data communication systems. Data compression may even be useful depending on the nature of the generated information.

Wireless E-mail, paging, system status information, system statistics and authorization systems are examples of short transactions systems. These users may all benefit from global mobile networks.

The proposed solution should therefore be able to support high transaction volumes with small portions of data exchanged during these transactions, support limited data display capabilities, support browsing capabilities and capabilities of linking to the Internet.

6.3 Proposed Solution

The proposed solution is illustrated in Figure 6-1. This protocol stack consists of the physical communication layer, the SRMA MAC layer, reduced version of AX.25, WDP, WTP, WSP and applications. The SRMA MAC layer is discussed in [18].

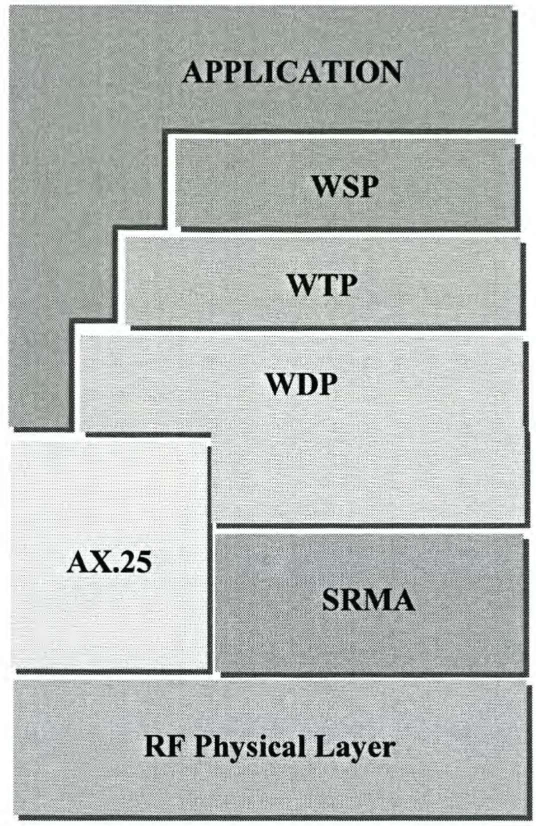

Figure 6-1 Proposed Architecture of field station

6.4 Transaction Orientated Data Exchange

WAP is designed to operate efficiently over narrow-band, high latency networks. WTP forms the foundation for the rest of WAP services. TCP transactions, commonly called T/TCP, an extension to TCP is designed for faster, more efficient and reliable client-server transactions[45]. TCP's three-way handshake is omitted at the beginning of a connection. T/TCP connection break-up is shortened as well. T/TCP has been optimized for traditional terrestrial networks where latency issues are not a factor. T/TCP's avoidance of the TCP's three-way handshake greatly benefits applications exchanging small amounts of data by reducing latency.

Measurements have shown that the **RTT** (**R**ound **T**rip **T**ime) for a terrestrial Internet link to be of the order of 80 ms. The RTT of a geostationary satellite has been measured to be approximately 560 ms[43]. Research is ongoing into various aspects of TCP implementation and performance of TCP/IP over satellite links. Issues include the slow start algorithm, the ability to accommodate large bandwidth-delay products, congestion control, acknowledgement and error recovery mechanisms.

Most of the research on Internet protocol performance over wireless links has focussed on TCP, the most commonly used transport protocol on the Internet. The main cause of the reported problems is the TCP assumption that all losses are due to congestion, so that loss detection triggers congestion avoidance techniques[56]. TCP/IPs Slow Start Algorithm is especially inefficient for relatively short data transfers compared to the delay-bandwidth product. Larger initial window sizes, byte counting, "spoofing" techniques, selective acknowledgements and other techniques are still under investigation.

SRMA favours exchange of relatively short bursts of data from mobile stations. WTP favours short transactions as well and SRMA acting as the underlying MAC layer would support enhanced performance. Both SRMA and WTP favour narrow bandwidth channels. Although WTP is a connectionless service, it contains reliable datagram delivery mechanisms.

6.4.1 Class 1 Transaction throughput

The maximum number of transactions that can be performed in an assigned timeslot is calculated in this and the following sections. The lower SRMA layer requests for a timeslot of length T seconds. The maximum number of WTP Class 1 transactions that may be performed during this interval of T seconds is calculated in the following paragraphs. The standard WTP packet format is used in the calculations.

8 bits	Protocol Identifier
16 bits	Destination Port
16 bits	Source Port
8 bits	Reference Number
8 bits	Total Segments
8 bits	Reserved for future extensions
D user bits	User data

Figure 6-2 WTP Frame structure

The WTP frame consists of 72+D bits. Data is transmitted at a rate of B bits/second. The transmission time for a WTP frame is therefore

$$\Gamma_{WTP} = \frac{72 + D}{B} \text{ seconds.}$$

The time the data travels from the mobile station to the satellite and vice versa is λ seconds. The time the satellite consumes to process the transaction frame is P_{SAT} seconds. The acknowledgement contains WTPack bits and its corresponding transmission time is

$$\Gamma_{ACK} = \frac{WTP_{ACK}}{B} \text{ seconds.}$$

Total time that is consumed for one Class 1 transaction to complete is

$$T_{class1WTP} = T_{WTP} + 2\lambda + P_{SAT} + T_{ACK} = \frac{72 + D}{B} + 2\lambda + P_{SAT} + \frac{WTP_{ACK}}{B} \text{ seconds.}$$

The maximum number of transactions that may be performed during the allocated time slot of length T seconds is

$$Total = \frac{T}{\Gamma_{class1WTP}} = \frac{BT}{72 + D + WTP_{ACK} + 2B\lambda + BP_{SAT}}$$

6.4.2 Class 1 HOLD-ON Transactions

The maximum number of transactions that may be executed during a predefined period of time is strongly dependant on the satellite processing time P. Should the satellite processing time be relatively long compared to the frame transmission and travelling time the total number of transactions will drop. With normal class 1 transactions, a next transaction is only performed when the previous one is completed. The number of transactions performed during a time-slot depends strongly on the satellite transaction processing time. Class 1 HOLD-ON transactions are useful in cases where multiple transactions may be processed at the same time. With Class 1 HOLD-ON transaction the mobile station transmits the request. When the satellite receives the requests, it issues a HOLD-ON acknowledgement to the mobile station. This indicates to the mobile station that the satellite has received its request and will return the result when the processing is completed. The satellite transmits the request when the processing is completed thereby completing the transaction. The duration of a class 1 HOLD-ON transaction is

$$T_{class1\,HoldOnWTP} = T_{WTP} + 3\lambda + P_{SAT} + \Gamma_{ACK} + \Gamma_{RESULT} = \frac{72 + D}{B} + 3\lambda + P_{SAT} + \frac{WTP_{HoldOnACK}}{B} + \frac{WTP_{RESULT}}{B}$$

The minimum time between two transactions is

$$\frac{72 + D}{B} + 2\lambda + \frac{WTP_{HoldOnACK}}{B}$$

The total number of class 1 transactions that may be performed during the assigned timeslot of T seconds is

$$Total = \frac{BT}{72 + D + 2B\lambda + WTP_{HoldOnACK}}$$

The absence the satellite transaction processing time result in a significant difference especially when long and complicated transactions are performed. It must be remembered that this total indicates only the total number of transactions that is initiated during the assigned timeslot. Since the downlink channel is exclusively reserved for satellite transmission, a result may be transmitted at a later stage after a field station's communication window has expired. The mobile stations therefore do not have to request a timeslot again since every mobile station in the footprint are capable of receiving the transmitted results.

6.4.3 Class 2 Transactions

A mobile station will issue an acknowledgement when it receives the result of a Class 2 transaction. To do this the mobile station must request for another time if one is not assigned to it at the time. Under worst case conditions the mobile stations must request for another timeslot, in addition to the original one that was used to initiate transactions. Acknowledgements of all the initiated transactions are transmitted in this timeslot. It is assumed that the mobile station waits until all the results for all the initiated transactions are returned. In the worst-case scenario all the initiated transactions must be acknowledged during the second timeslot. The total number of transactions that must be acknowledged by the mobile station is equal to the total number of class 1 transactions. This additional request for another timeslot adds a significant delay to the throughput of a mobile station.

6.4.4 GES Server

In the previous sections the satellite is acting as the server. All transactions are performed by the satellite. In the following scenario, the satellite will only act as a relay between the mobile station and the GES. The GES will now act as transaction server instead of the satellite. This has a number of advantages namely

- The GES has access to greater number of resources. The GES is far more flexible than the satellite in terms of power and processing requirements. A greater variety of transactions may be performed by the GES.
- A GES connected to a network may even access other resources on other networked devices to complete a transaction.
- It is assumed that the GES has a dedicated channel with the satellite while it is in the satellites footprint. The GES therefore do not have to request for a timeslot.

When the GES is used an extra hop is introduced. This amounts to an extra delay of λ for a message travelling from the mobile station to the satellite and from the satellite travelling to the GES. The time taken to complete a Class 1 transaction changes now to

$$\mathrm{T}_{class1WTP} = \mathrm{T}_{WTP} + 4\lambda + \mathrm{P}_{GES} + \mathrm{T}_{ACK} = \frac{72 + D}{B} + 4\lambda + \mathrm{P}_{GES} + \frac{WTP_{ACK}}{\mathrm{B}} \text{ seconds}$$

where P_{GES} represents the transaction processing time of the GES. If the Class 1 transaction processed on a GES is to be shorter than the same class 1 transaction processed on the satellite the following equation must be satisfied

$$4\lambda + P_{GES} < 2\lambda + P_{SAT}$$

If P_{SAT} is expressed in terms of P_{GES} as follows

$$P_{SAT} = \Omega P_{GES} \text{ then}$$

$$4\lambda + \Omega P_{SAT} < 2\lambda + P_{SAT}$$

$$(1 - \Omega)P_{SAT} > 2\lambda$$

$$(1 - \Omega) > \frac{2\lambda}{P_{SAT}}$$

$$\Omega < 1 - \frac{2\lambda}{P_{SAT}}$$

If the satellite transaction time is shorter than 2λ then the satellite server will always be able to process more transactions than the GES.

6.5 Session Services

WSP must provide robust services to applications. The GSMFTP implementation on GSM mobile stations support robust operation when portions of data are corrupted, connections are lost due to bad signals or because mobile is moving out of the GSM coverage area. A similar FTP with similar requirements must be developed for such a platform. Such a FTP application will allow mobile stations to start the file transfer during one satellite pass, suspend the service when the mobile station moves from the satellites communication footprint, resume and finish the service with a next satellite pass. Such an FTP application may, under severe noise and congestion conditions, continue to execute the file transfer using several satellite passes. Like the GSMFTP protocol, the file transfer procedure must contain enough information for it to be allowed to continue.

The ultimate FTP procedure would allow a mobile subscriber to transfer portions of a file to different satellites in a constellation. Several different gateways may then download the different file segments from the satellites. These gateway(s) must then be able to reconcile the different file segments into one file that matches the original file. GSMFTP contains all the elements to support these features in a mobile environment. WSP contains the facilities for a server to suspend a session with a particular client. The client may resume the same session with the same or with another server.

6.6 Security Layer

As e-commerce increases, the need for reliable Internet security does as well. Potential of interception and corruption may be increased by the wide-area coverage of satellite links. **IPSec** (**I**nternet **P**rotocol **Sec**urity) mechanisms must ensure confidentiality, authentication, integrity, access control and key management.

Confidentiality means that only the appropriate users have access to the information. Authentication requires verifications of a user's identity and right to access. Integrity means that information has not been modified or corrupted. Access control ensures that the system cannot be compromised by unauthorised access.

HTTP (**H**yper**T**ext **T**ransfer **P**rotocol) forms the foundation of the *World Wide Web*. In a typical HTTP client-server transaction the client does the active open, sends a small request to the server, the server sends back a reply and the server closes the connection. By effectively choosing WTP on SRMA these client-server transactions may be processed more efficiently.

6.7 Application Layer functionality

Features that may be inherited from the three different implementations are:

- ORBCOMM speed dial functionality that is useful to indicate the destination of the data.

- WTP frames which is based on transaction-orientated data transfer which is useful for short bursts of data, as in the case of the SUNSAT mobile field station.

- WTP and transaction orientated data transfers support browsing functionality such list the files in a given directory, list the processes running on the satellite, change a configuration setting on the satellite, download a file, upload a file, etc.

- WTP has the option to send data using one of three available transaction classes. As in the case of ORBCOMM, different message types are available for transmission of data under different circumstances. The three different WTP transactions may be used in different circumstances as well pertaining to the requirement support needed and the certainty of correct data delivery.

- WTP assists with effective communication in a LEO communication environment where noisy channels, sudden and temporary loss of communication channels because of natural or man-made structures and multi-path propagation.

- Multiple application services on the field station may now communicate via a single timeslot. WDP allows for communication of multiple instances over a single lower bearer service. A large data transfer and a number of small transactions may be in progress using the assigned timeslot.

- GSMFTP transfers large portions of data. WSP assists GSMFTP to suspend the data transfer procedure and the session during a temporary loss of communication or when SUNSAT is no longer visible to the mobile field station. WSP and GSMFTP may be resumed again when communication with SUNSAT is possible.

- GSMFTP and WSP support transfer of large amounts of data using low-bandwidth channels. The restart mechanisms in GSMFTP allows for a data session to spand several satellite passes.

- WSP uses transaction-based data transmission, in the same fashion as T/TCP, which was found to be more efficient in request-response browsing operations.

- Out-of-coverage queues similar to the one implemented in the GSM Fleet Management system are implemented in the managing entity of the WAP layers.

- The WDP may act as multiplexer, in the same fashion as the DXN layer in the DX Protocol stack, that allows for communication over one or more available bearer services. Communication may be done using AX.25 and SRMA without the upper communication layers being aware of it.

6.8 Conclusion

The capability of the mobile field station is expanded with the addition of the extra software middleware layers expand the mobile field station capability. The SRMA is a MAC Protocol Layer and cannot therefore provide enough functionality to cover a wide range of applications. The inclusion of the WAP WDP, WTP and WSP layers allows the mobile station to take advantage of high volumes of short transaction that may be performed in one assigned timeslot, the use of request-response transactions that can easily support browsing functionality and session suspend and resume facilities. Session suspend and resume facilities allow the field station together with the GSMFTP protocol can be used to transfer large portions of data over a number of different satellite passes without the upper layer applications being aware of it. This is a truely powerful feature that allows a field station to literally transfer a large single data file over a period of a number of days. Only a small period per satellite pass is therefore needed to resume a suspend and suspend a FTP session until data transfer is complete.

Chapter 7

Conclusion

The SUNSAT SRMA protocol is a MAC layer protocol that regulates and shares a single up-link channel to SUNSAT among multiple field stations. SRMA is sufficient to control and regulate use of the up-link channel, but is insufficient to effectively combat LEO propagation factors such as noisy and poor communication channels, multi-path propagation observed at low satellite elevation angles and temporary termination of data transfer procedures when the satellite is not visible to the field station anymore. Additional functionality is therefore needed to allow for data retransmission, transmission of short and long data bursts, graceful termination when satellite becomes invisible while transferring data between field station and satellite, and the re-start of terminated data transfers.

Three wireless communication systems were successfully investigated, designed and implemented. The DataTREK Data Collection system is a prime example of the type of applications that the SUNSAT field station should be able to support. The GSM fleet management system operates in a store-and-forward fashion as well as using circuit-switched data connections to transfer large amounts of data. The robust GSMFTP protocol is designed to operate in poor communication conditions and contains an explicit re-start mechanism to continue data transfer procedures that were terminated earlier on. WAP techniques and procedures were for possible inclusion into the SUNSAT field station design that would allow for efficient communication in low-bandwidth data channels.

This thesis proposes the introduction of additional layers of software, called middleware, that supports process to process communication, interactive browsing of SUNSAT's file systems, and client-server transactions. Transaction-based communication, utilised in WAP's WTP layer, proved to be an effective method of communication especially when short client-server request-response operations must be supported WTP. SRMA is geared towards small bursts of data. WTP transaction-based communication supports this paradigm since no specific channel setup and tear-down procedures are used. These connection setup and tear-down procedures are responsible for most of the overhead involved in transferring a small amount of data using connection-orientated services such as AX.25 and PACSATs FTP1. It therefore ultimately supports high-transaction volumes with low-overhead.

The GSMFTP protocol, Out-of-Coverage queues and the multiplexing DXN layer used in the DX Protocol Stack, support communication in a LEO store-and-forward and connection-orientated communication environment. The GSMFTP contains an ACK-N acknowledgement scheme that acknowledges a fixed number of PDUs received. This scheme is simple to implement, contains a rudimentary flow-control scheme embedded in its design and allows for easy implementation of the required restart procedure.

Different network architectures, regarding the integration of WAP into LEO satellites were investigated and proposed. ORBCOMM subscribers may benefit by the introduction of rudimentary WAP WTP functionality. A WAPSAT is a LEO store-and-forward satellite that uses the WAP Protocol Stack to regulate satellite-to-field-station communication. SUNSAT may operate as a WAPSAT as well when using WAP compatible field stations. Integration of WAP, DX Protocol and ORBCOMM features into the SUNSAT field station would certainly allow for flexible and effective transfer of short and long data bursts in the store-and-forward LEO communication environment

References

[1] Ackerman, M.C. "*Kernel Support for Embedded Reactive Systems*", M.Sc Thesis, University of Stellenbosch, October 1993.

[2] Allery M.N., Price H.E., Ward J.W., Da Silva Curiel R. A. "Low Earth Orbit micro-satellites for data communications using small terminals", Tenth International Conference on Digital Satellite Communications, Vol. 2, 15-19 May 1995, pp. 457-465.

[3] Ananasso F., "An overview of mobile satellite systems and their evolution towards satellite personal communication networks", Tenth International Conference on Digital Satellite Communications, Vol. 2, 15-19 May 1995, pp. 435-440.

[4] Becerra A., et al "Service Definition in the Integrated Satellite UMTS Environment", Tenth International Conference on Digital Satellite Communication, Vol. 2, 15-19 May 1995, pp. 442-449.

[5] Cardoza A.R. *"A GSM FTP Protocol for Remote LEO Gateways",* 2nd Annual Proceedings of the South African Telecommunications Networks and Applications Conference, September 1999, pp 270-274.

[6] Cardoza A.R. "*DX50 Protocol Stack Specification*", FTP1C06, ESL, University of Stellenbosch, 31 March 2000.

[7] Cardoza A.R., Mostert S. "*Turning PACSATs into WAPSATs*", SATNAC 2000, August 2000.

[8] Cardoza A.R., Mostert S. "*WAP via ORBCOMM*", AIAA Utah State University Conference on Small Satellites, Utah State University, 2000.

[9] Beech W.A., Nielsen D.E., Taylor J. "*AX.25 Link Access Protocol for Amateur Packet Radio*", Version 2.2, Tucson Amateur Packet Radio Corporation, 11 November 1997.

[10] Choon Chan M., Woo T.Y.C. "Next-Generation Wireless Data Services: Architecture and Experience", IEEE Personal Communications, February 1999, pp. 20-29.

[11] Connors D.P., Ryu B., Dao S. "Modeling and Simulation of Broadband Satellite Networks. Part 1: Medium Access Control for QoS Provisioning", IEEE Communications Magazine, March 1999, pp. 72-79.

[12] Evans, J.V. "Satellite Systems for Personal Communications", IEEE Antennas and Propagation Magazine, Vol. 39, No. 3, June 1997, pp. 7-19.

[13] Faergemand O., Olsen A. "*Introduction to SDL-92*", TFL, 27 July 1992.

[14] Fasbender A., Reichert F. "Any Network, Any Terminal, Anywhere", IEEE Personal Communications Magazine, April 1999, pp. 23-30.

[15] Garg V.K., Smolik K.F., Wilkes J.E. "Applications of CDMA in Wireless/Personal Communications", Prentice Hall PTR, NJ.

[16] Gavilan J., Beccera A., Berberana I. "LAN Internetworking using VSAT Systems", Tenth International Conference on Digital Satellite Communications, 15-19 May 1995, Vol. 2, pp. 491-496.

[17] Ghani N., Dixit S. "TCP/IP Enhancements for Satellite Networks", IEEE Communications Magazine, July 1999, pp. 64-72.

[18] Grobler F., "Design and implementation of a store-and-forward communication system on the SUNSAT microsatellite", M.Sc Engineering Thesis, University of Stellenbosch, October 1996.

[19] Grobler H., Cardoza A. "*Mobile and Base Station Protocol Stack Implementation*", FTP1A03, ESL, University of Stellenbosch, 21 May 1999.

[20] GSM 07.05, "GSM: Digital Cellular Telecommunications System (Phase 2+); Use of Data Terminal Equipment – Data Circuit terminating; Equipment (DTE-DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS)", Version 5.5.0, ETSI, January 1998.

[21] GSM 07.07, "GSM: Digital cellular telecommunications system (Phase 2+); AT Command set for GSM Mobile Equipment (ME)", Version 5.0.0, ETSI, July 1996.

[22] Kaasinen E., Aaltonen M., et al, "Two approaches to bringing Internet services to WAP devices", Computer Networks, Vol 33, 2000, pp. 231-246.

[23] Lettieri P., Srivastava M. B. "Advances in Wireless Terminals", IEEE Personal Communications Magazine, February 1999, pp. 6-18.

[24] Marzoli A., Ruspantini L., Scagnoli R., "Mobile Satellite Communication Systems: A Competitive scenario for new systems and services", Tenth International Conference on Digital Satellite Communications, Vol 2, 15-19 May 1995, pp. 450-455.

[25] Mehrotra A. "GSM System Engineering", Artech House Inc., MA, 1997.

[26] Montgomery J. "The Orbiting INTERNET: Fiber in the Sky", Byte Magazine, November 1997, pp. 58-72.

[27] Mostert S., Cardoza A. "*Remote measurement and control via satellite constellation*", Elektron Magazine, November-December 1999, p 6.

[28] Mostert S., Grobler H., Cardoza A. "*Proposal for Base Station Interface*", FTP1A01, ESL, University of Stellenbosch, 13 May 1999.

[29] Motorola, Inc. "*d10 Integrated Wireless Modem: Integration and Application Developers Guide*", Document No. 6804331L14-O, Draft August 1998.

[30] ORBCOMM Global, L. P. *"ORBCOMM Gateway Customer Access Interface Specification*", Revision B – D20050006, 6 January 1998.

[31] ORBCOMM Global, L. P. *"ORBCOMM Serial Interface Specification"*, Revision C – E80050015, 30 June 1998.

[32] ORBCOMM Global, L. P. *"ORBCOMM System Overview",* Revision B – A80TD0008, 26 April 1998.

[33] Ors T., Sun Z., Evans B.G. "A MAC Protocol for ATM over Satellite", Telecommunications 1998, 29 March – 1 April 1998, Conference Publication No. 451, pp. 185-189.

[34] PC/104 Consortium, *"PC/104 Specification ",* Version 2.3, June 1996.

[35] Peyravi H. "Medium Access Control Protocols Performance in Satellite Communications", IEEE Communications Magazine, March 1999, pp. 62-71.

[36] Philips K.P. "An overview of propagation factors influencing the design of mobile satellite communication systems", 10th International Conference on Antennas and Propagation, 14-17 April 1997, Conference Publication No. 436, pp. 261-268.

[37] Price H.E., Ward J. "*PACSAT Protocol Suite- An overview*", ARRL 9th Computer Networking Conference, August 1990.

[38] Price H.E., Ward J. "*PACSAT Broadcast Protocol*", ARRL 9th Computer Networking Conference, August 1990, pp.232-238.

[39] RACAL Electronics South Africa, "Application Level Interface Definition Document for the TRACS GSM SMS Protocol", Document No. RESA-00-010-IDD, Issue 8, 1 April 1999.

[40] Redl S.M., Weber M.K. , Oliphant M.W. "An Introduction to GSM", Artech House Inc., Boston, 1995.

[41] Retief F., "*Implementation of a AX.25 protocol stack on a 8-bit micro-controller"*, Thesis Project 448, University of Stellenbosch, November 1999.

[42] Salkintzis A.PK. "Packet Data Over Cellular Networks: The CDPD Approach", IEEE Communications Magazine, June 1999, pp. 152-159.

[43] Selmer R.M. "Internet via Satellite", 2nd Annual Proceedings of the South African Telecommunications Networks and Applications Conference, September 1999, pp. 333-337.

[44] Stellar Satellite Communications Ltd., *"Enhanced ORBCOMM Serial Interface for Stellar Terminals"*, Doc No. 0517638, April 1999.

[45] Stevens W.R. "TCP/IP Illustrated, Volume 3", Addison-Wesley Professional Computing Series, 1996.

[46] Taaghol P., Evans B.G. "Satellite UMTS/IMT2000 W-CDMA Air Interfaces", IEEE Communications Magazine, September 1999, pp. 116-126.

[47] Turletti T., Tennenhouse D. "Complexity of a Software GSM Base Station", IEEE Communications Magazine, February 1999, pp. 113-117.

[48] Van Nobelen R., Seshadri N., Whitehead J., Timiri S. "An Adaptive Radio Link Protocol with Enhanced Data Rates for GSM Evolution", IEEE Personal Communications Magazine, February 1999, pp. 54-64.

[49] Varshney U., Vetter R., "*Emerging Mobile and Wireless Networks*", Communications of the ACM, Vol. 43, No. 6, June 2000, pp. 73-81.

[50] WAP Forum, "*Wireless Application Protocol: Architecture Specification*", http://www.wapforum.org, Version 30 April 1998.

[51] WAP Forum, "*Wireless Application Protocol: Wireless Session Protocol Specification*", http://www.wapforum.org, Version 5 November 1999.

[52] WAP Forum, "*Wireless Application Protocol: Wireless Transaction Protocol Specification*", http://www.wapforum.org, Version 11 June 1999.

[53] WAP Forum, "*Wireless Application Protocol: Wireless Datagram Protocol Specification*", http://www.wapforum.org, Version 05 November 1999.

[54] Ward J., Price H.E. "*PACSAT Protocol: File Transfer Level 0*", ARRL 9th Computer Networking Conference, August 1990, pp. 209-231.

[55] Wood L., Cruickshank H., Sun Z. "Support group applications via Satellite constellations with multicast", Telecommunications 1998, Conference Publication No. 451, 29 March-1 April 1998, pp 190-194.

[56] Xylomenos G., Polyzos G.C. "Internet Protocol Performance over Networks with Wireless Links", IEEE Network, July/August 1999, pp. 55-63.

[57] Zorzi M., Rao R.R. "Perspectives on the Impact of Error Statistics on Protocols of Wireless Networks", IEEE Personal Communications Magazine, October 1999, pp. 32-40.

Appendix A
SDL Charts

A.1 DX Protocol System Overview

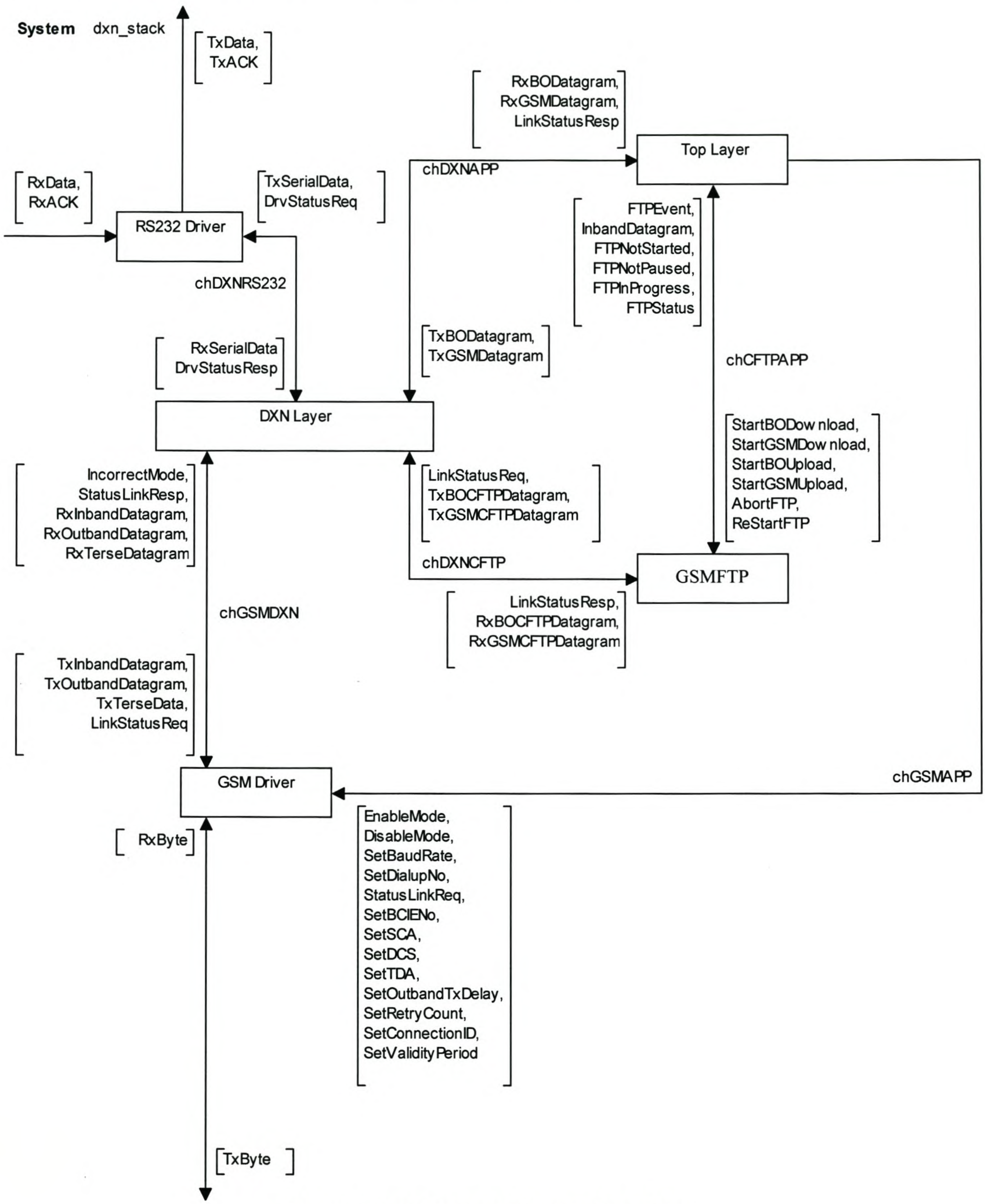

Figure A.1 DX Protocol System Overview

A.1.1 GSM Device Driver

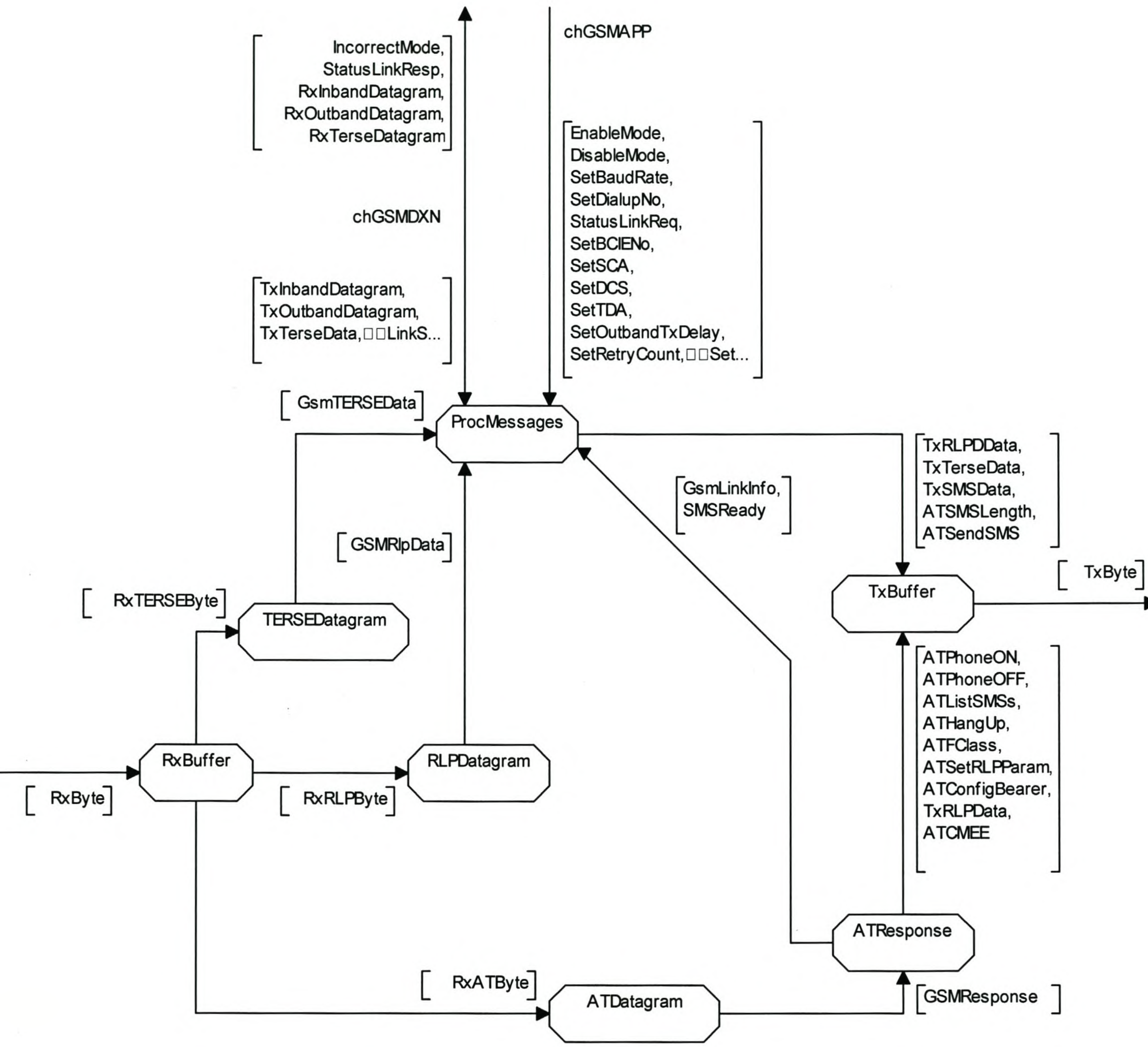

Figure A.2 GSM Driver Processes

A.1.1.1 RxBuffer Process

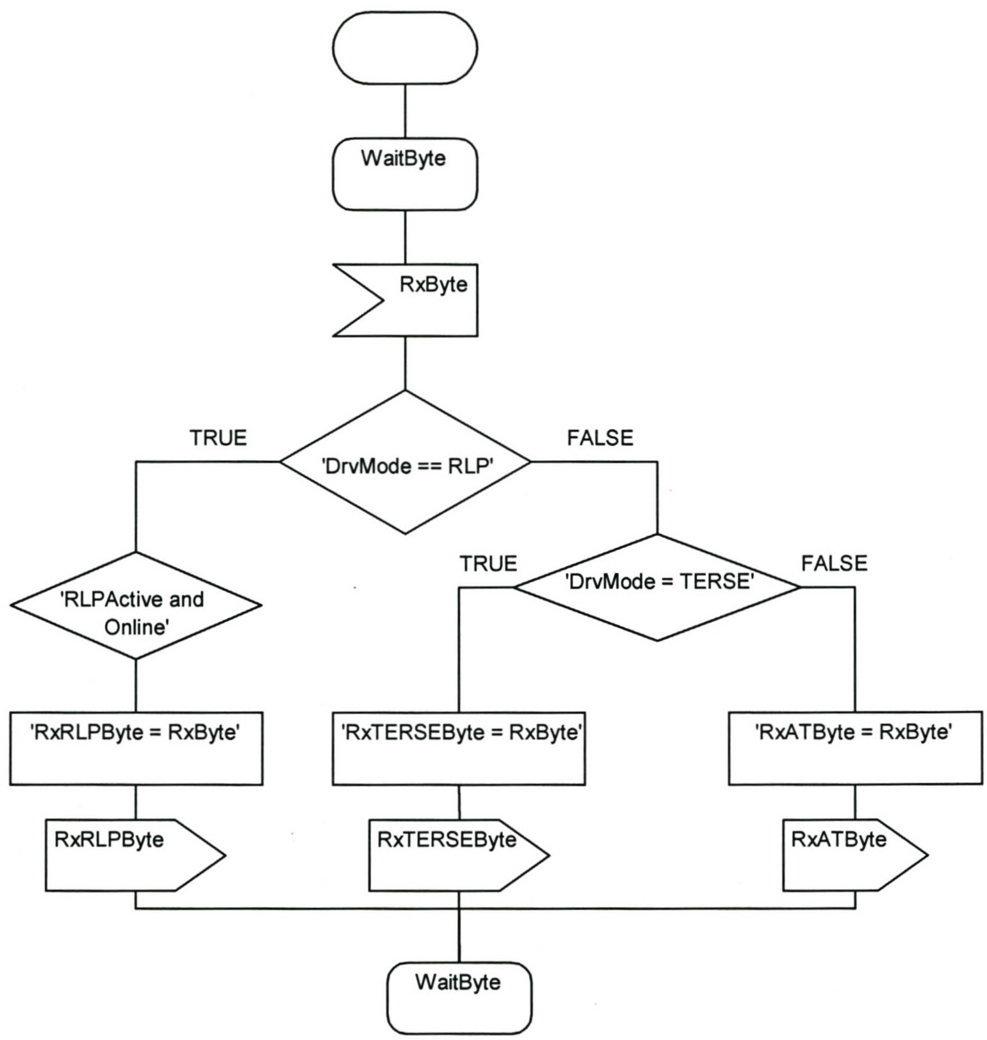

Figure A.3 GSM Driver's RxBuffer Process

A.1.1.2 TERSE Datagram Process

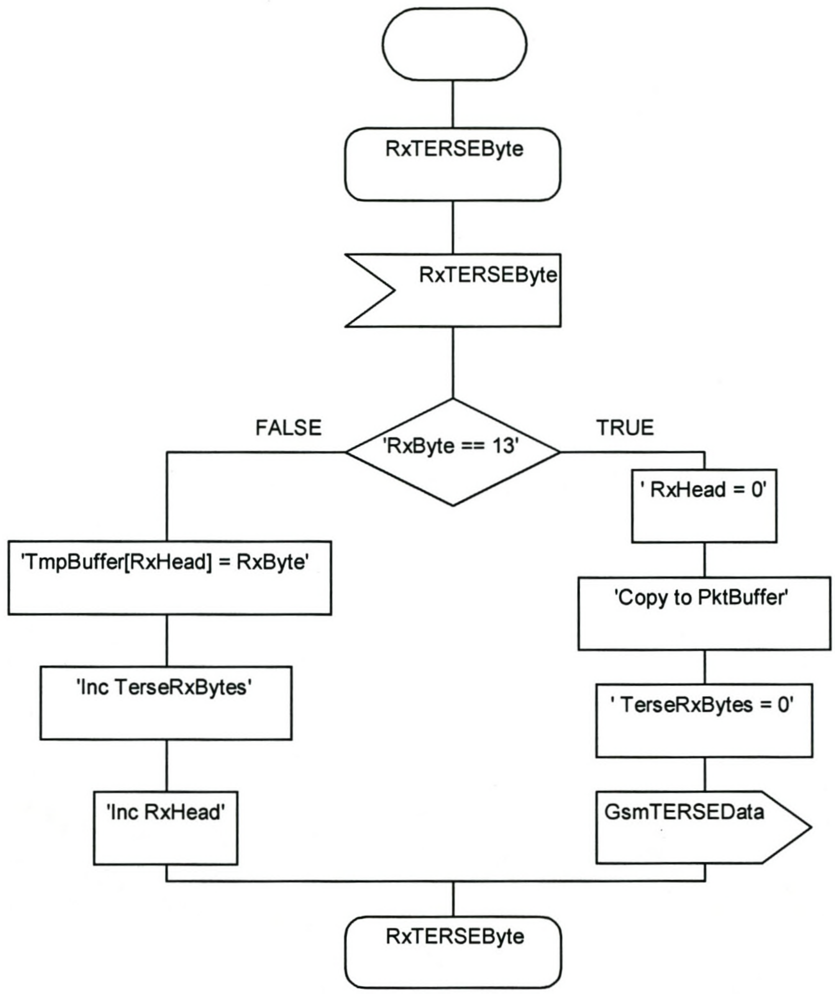

Figure A. 4 GSM Driver's TERSE Data Process

A.1.1.3 AT Datagram Process

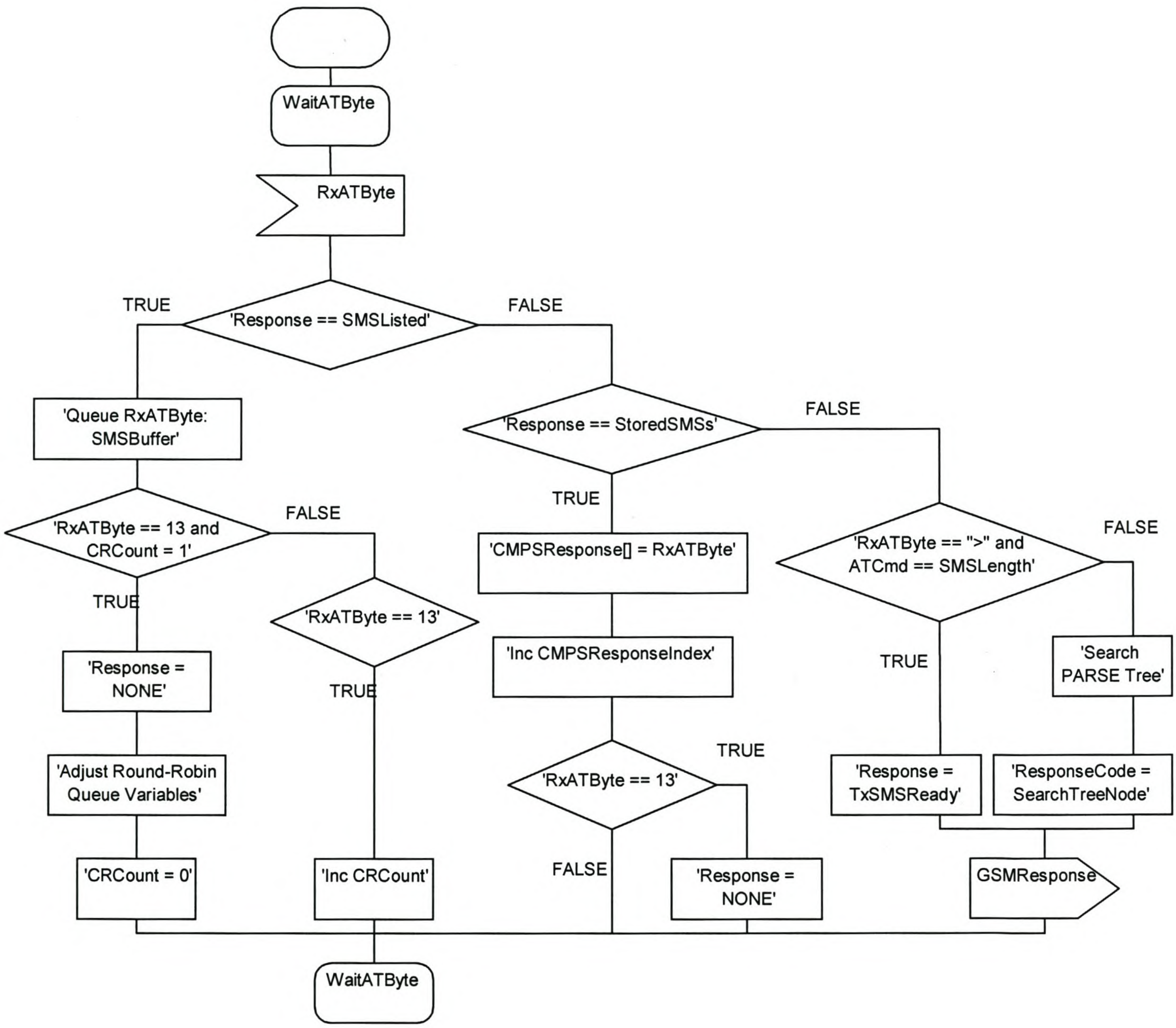

Figure A.5 GSM Driver's AT Datagram Process

A.1.1.4 RLP Datagram Process

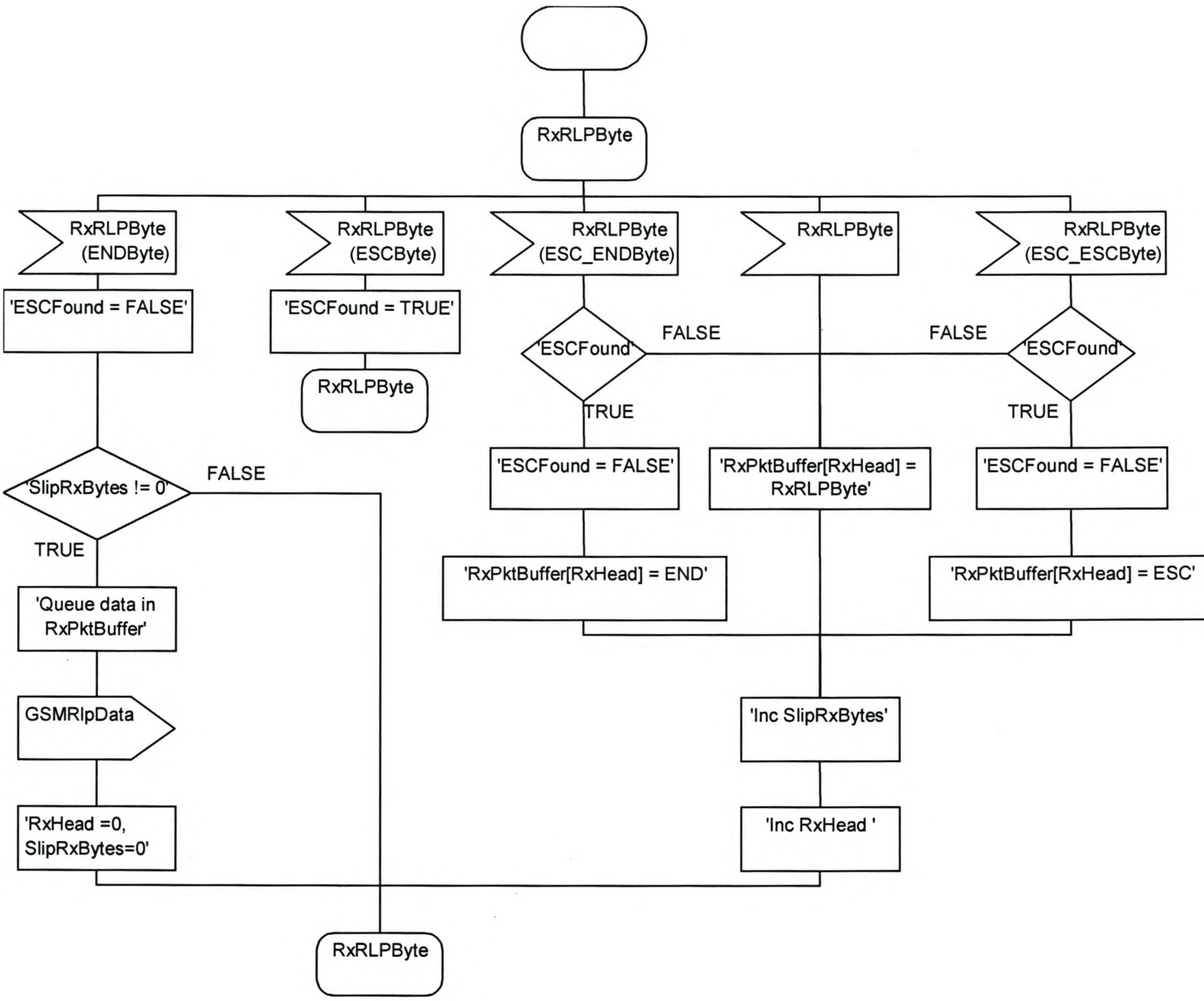

Figure A. 6 GSM Driver's RLP Datagram Process

A.1.1.5 AT Response Process

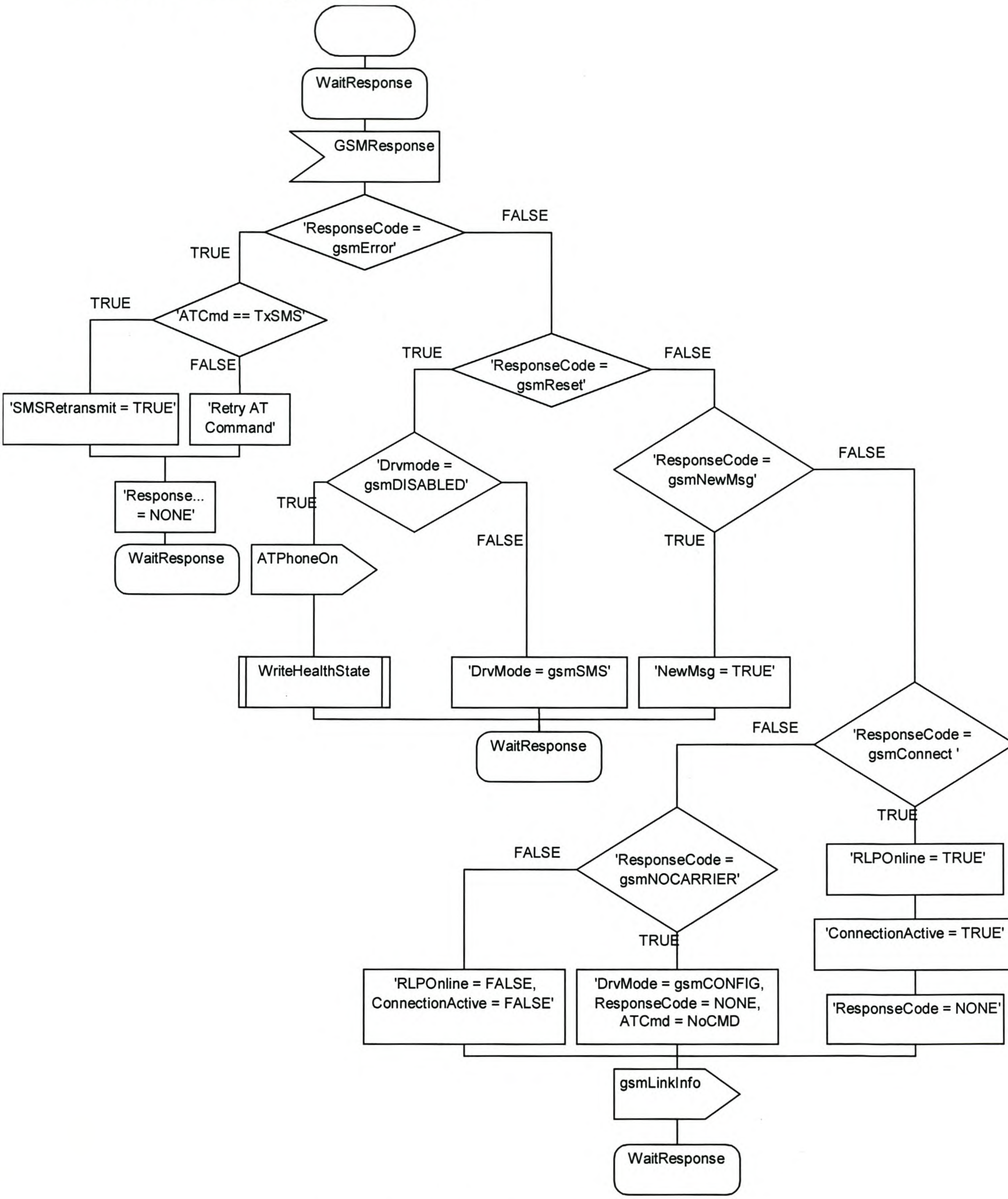

Figure A.7 GSM Driver's AT Response Process

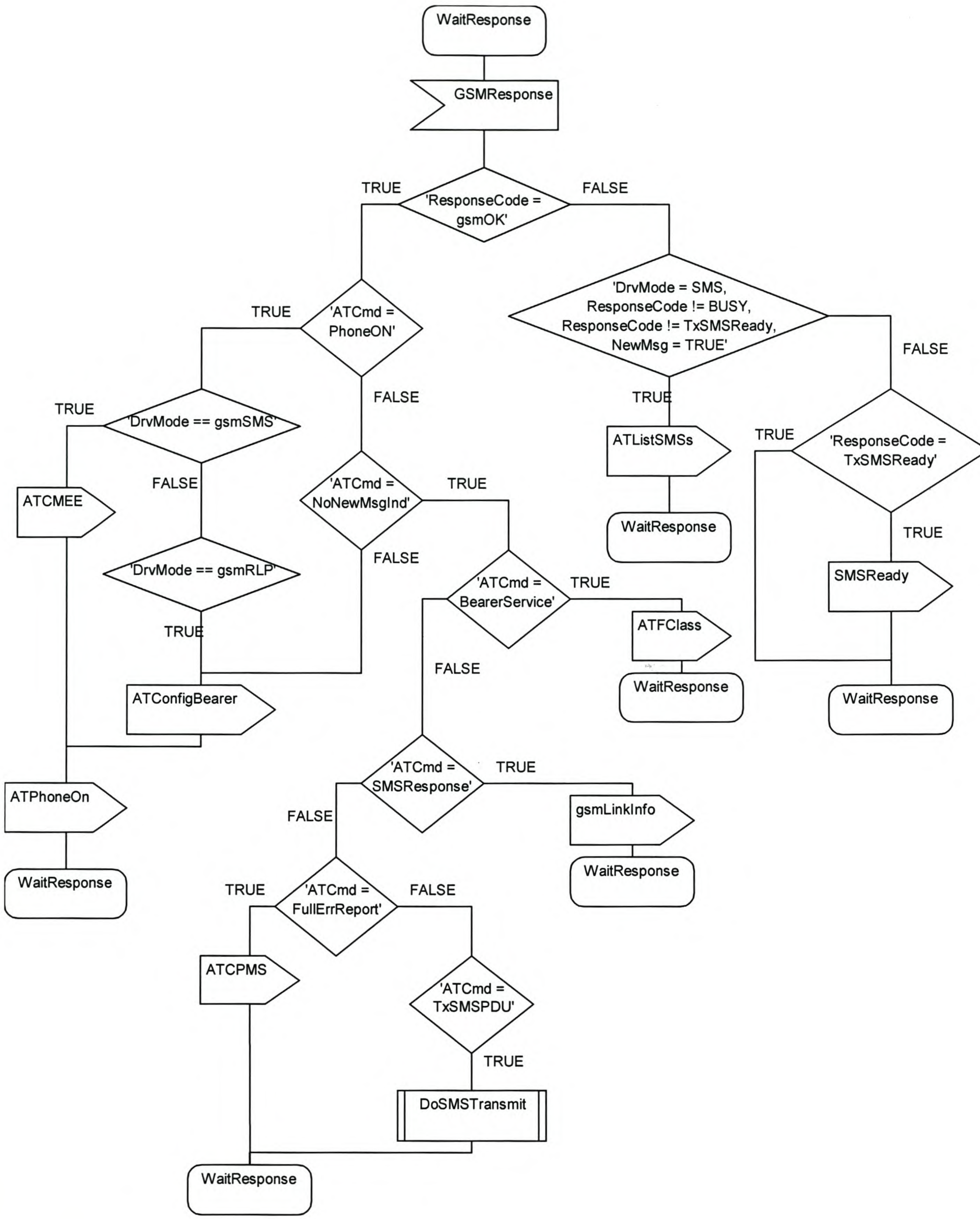

Figure A.8 GSM Driver's AT Response Process

A.1.1.6 ProcMessages Process

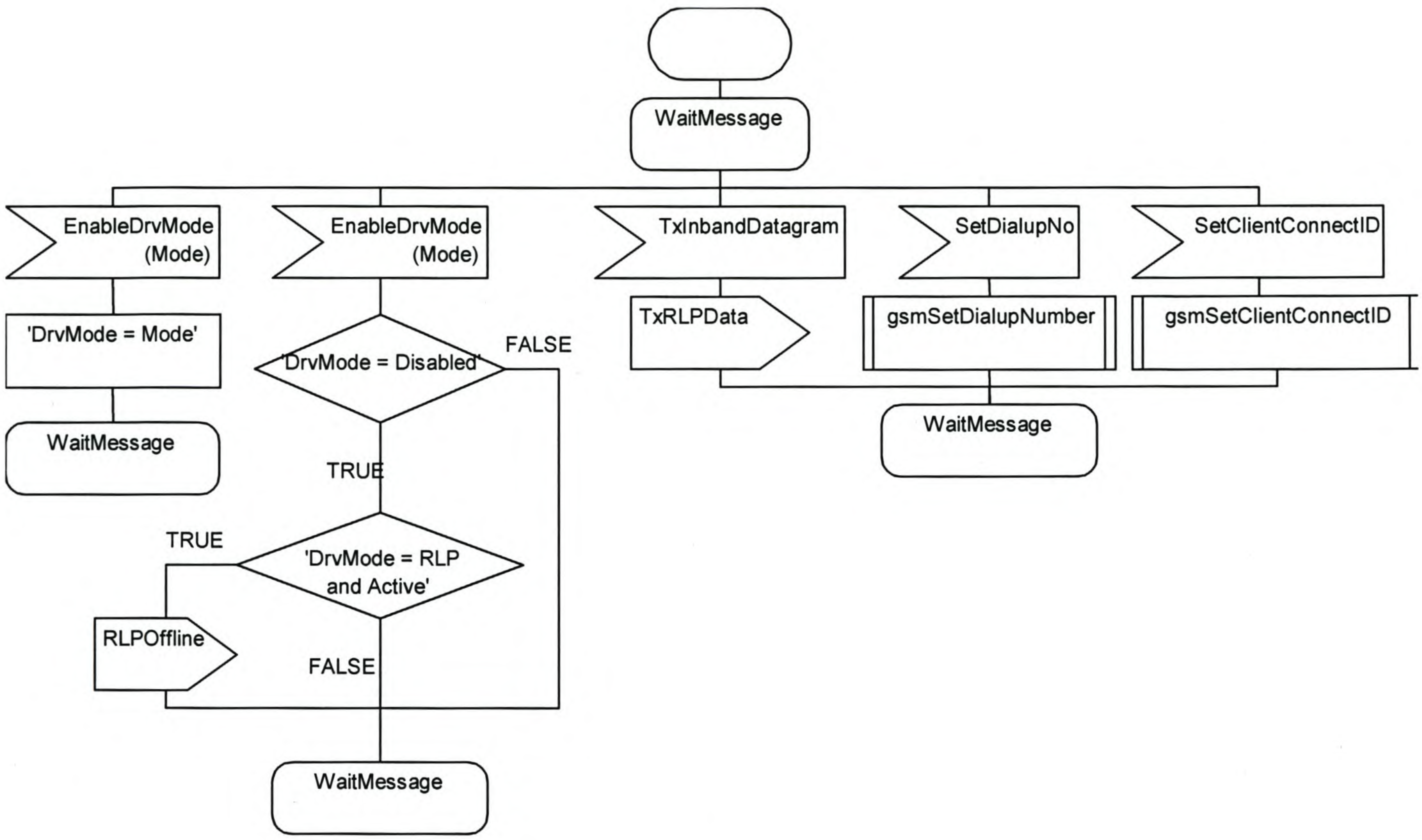

Figure A.9 GSM Driver's ProcMessages Process

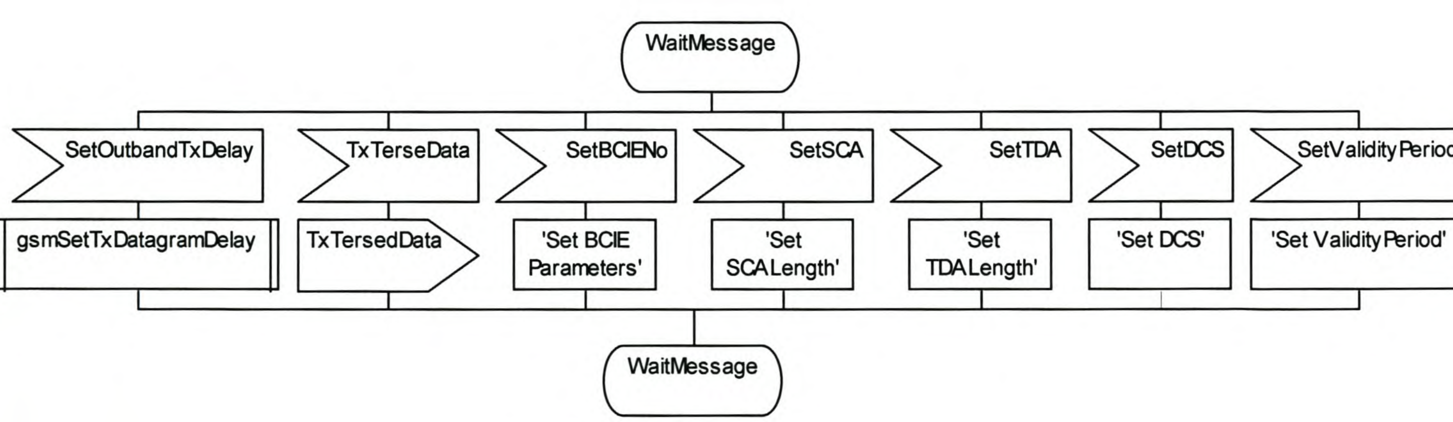

Figure A.10 Driver's ProcMessages Process

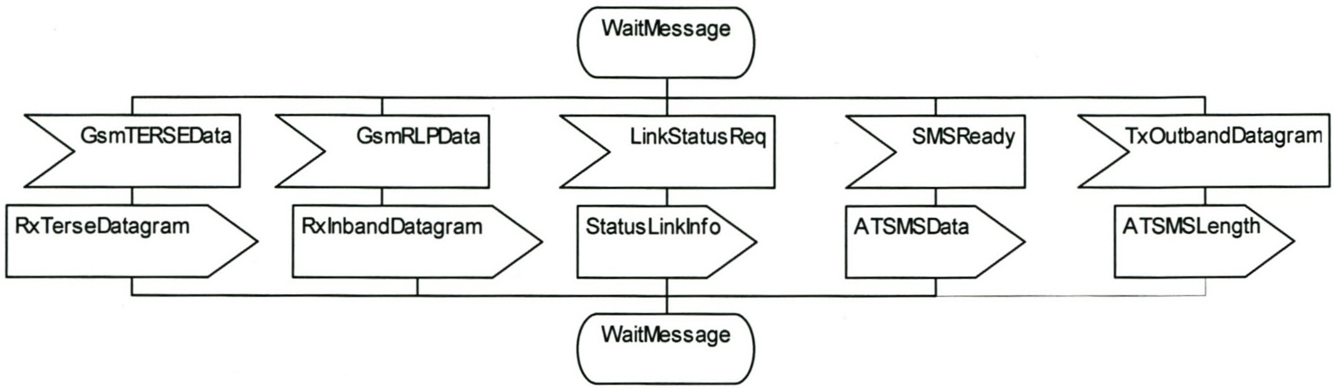

Figure A. 11 GSM Driver's ProcMessages Process

A.1.2 DXN Layer

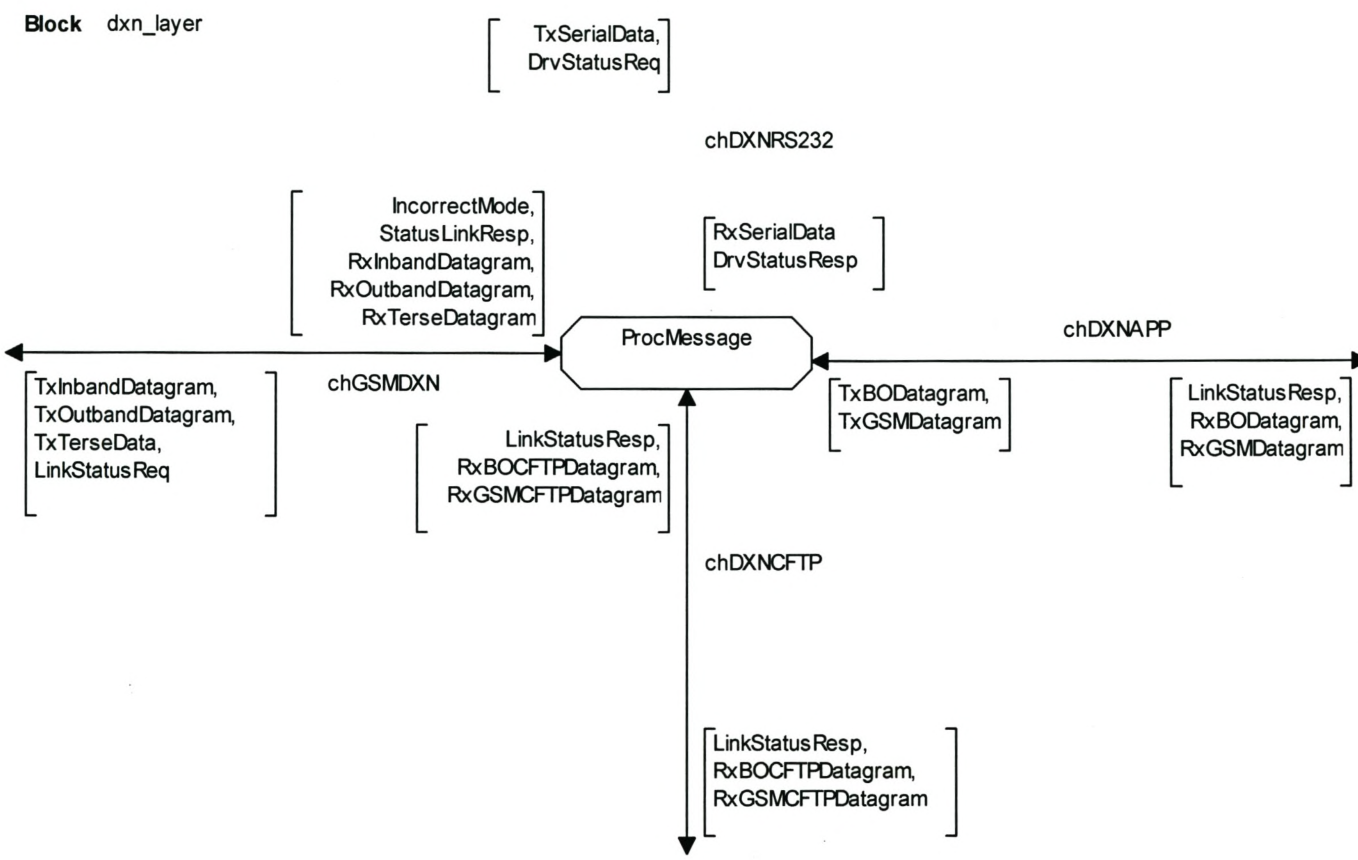

Figure A.12 DXN Protocol Process

A.1.2.1 DXN ProcMessage Process

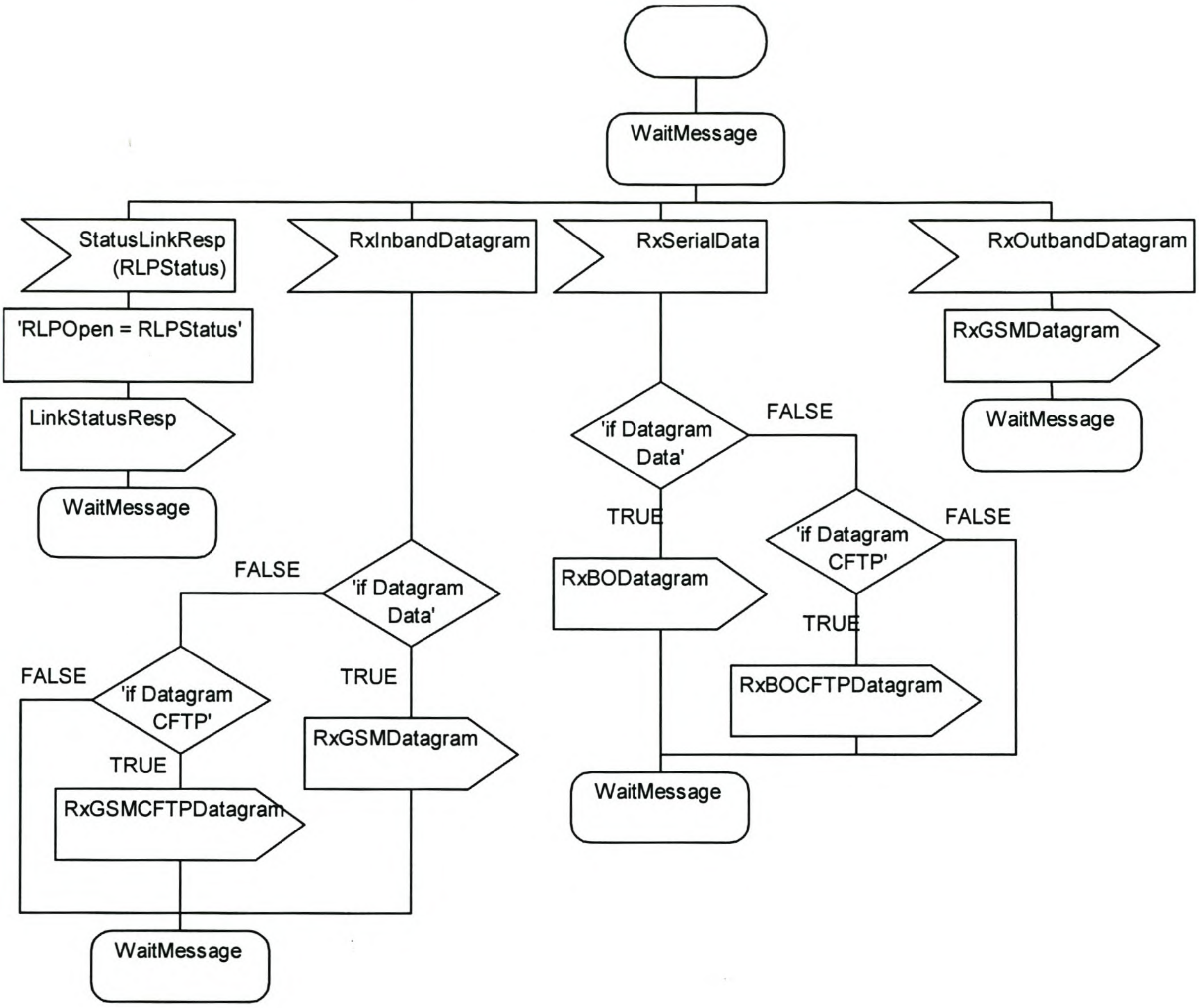

Figure A.13 DXN ProcMessage Process

A.1.3 GSMFTP Protocol Layer

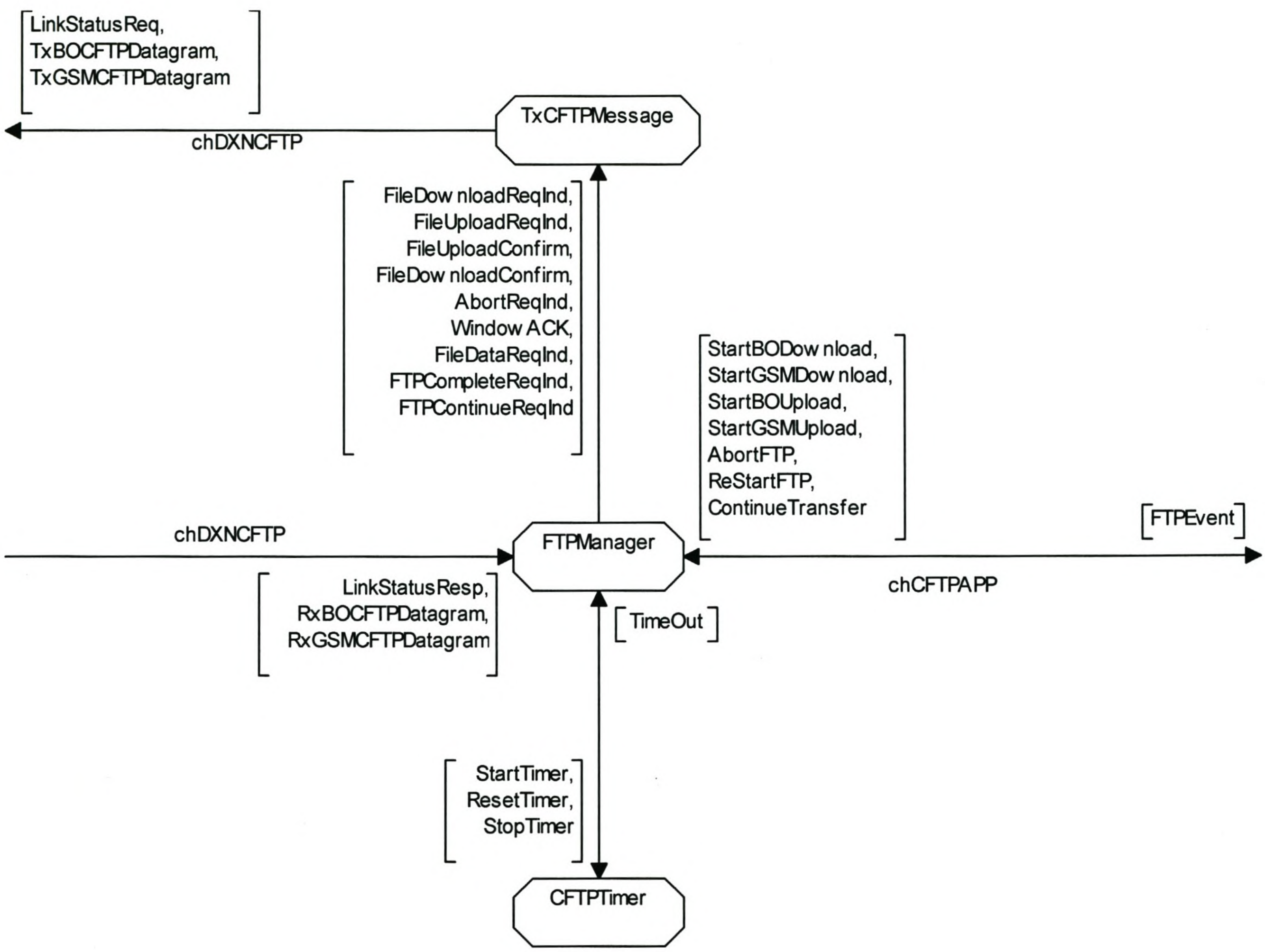

Figure A.14 CFTP Process

A.1.3.1 FTP Manager Process

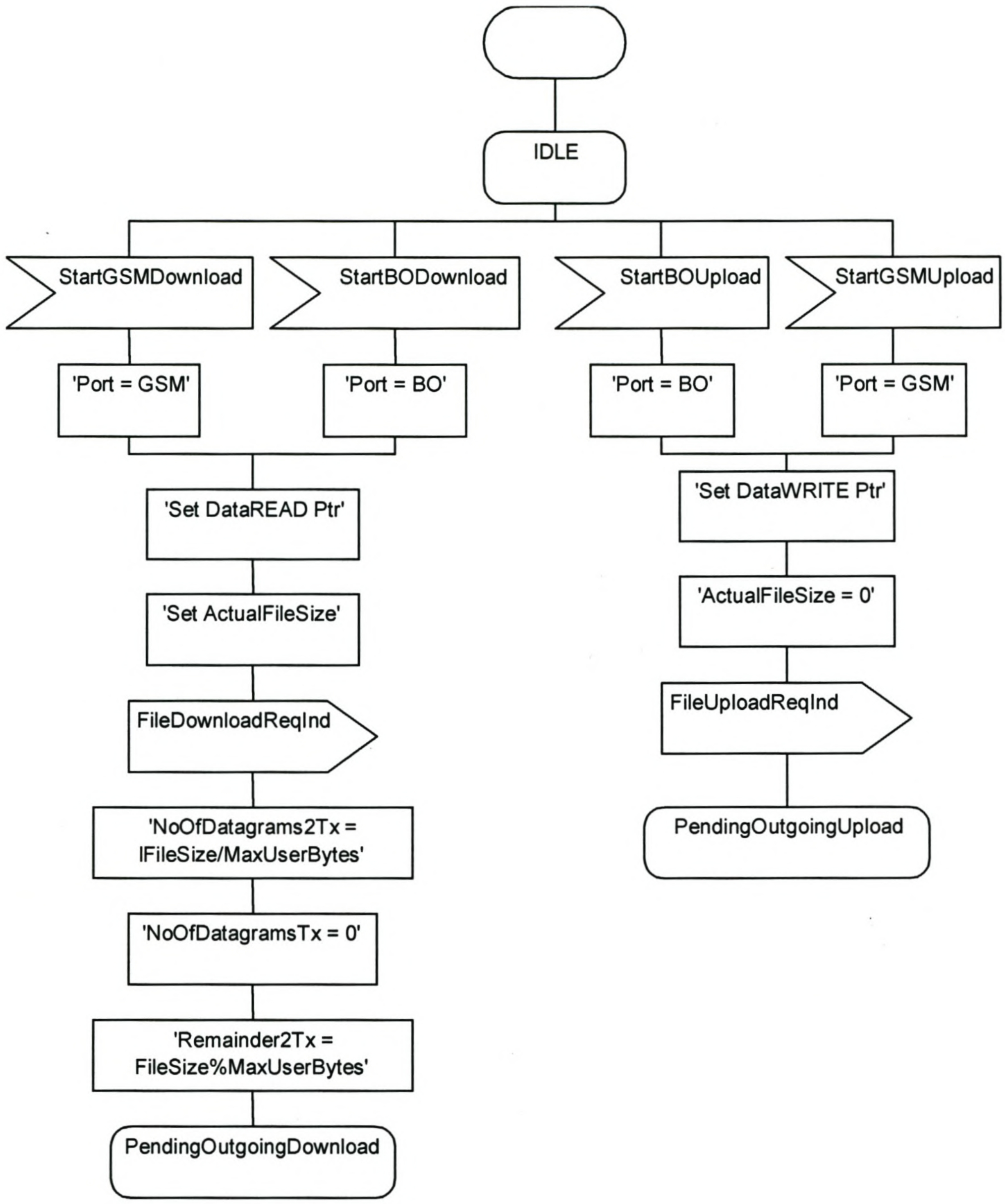

Figure A.15 FTP Manager Process

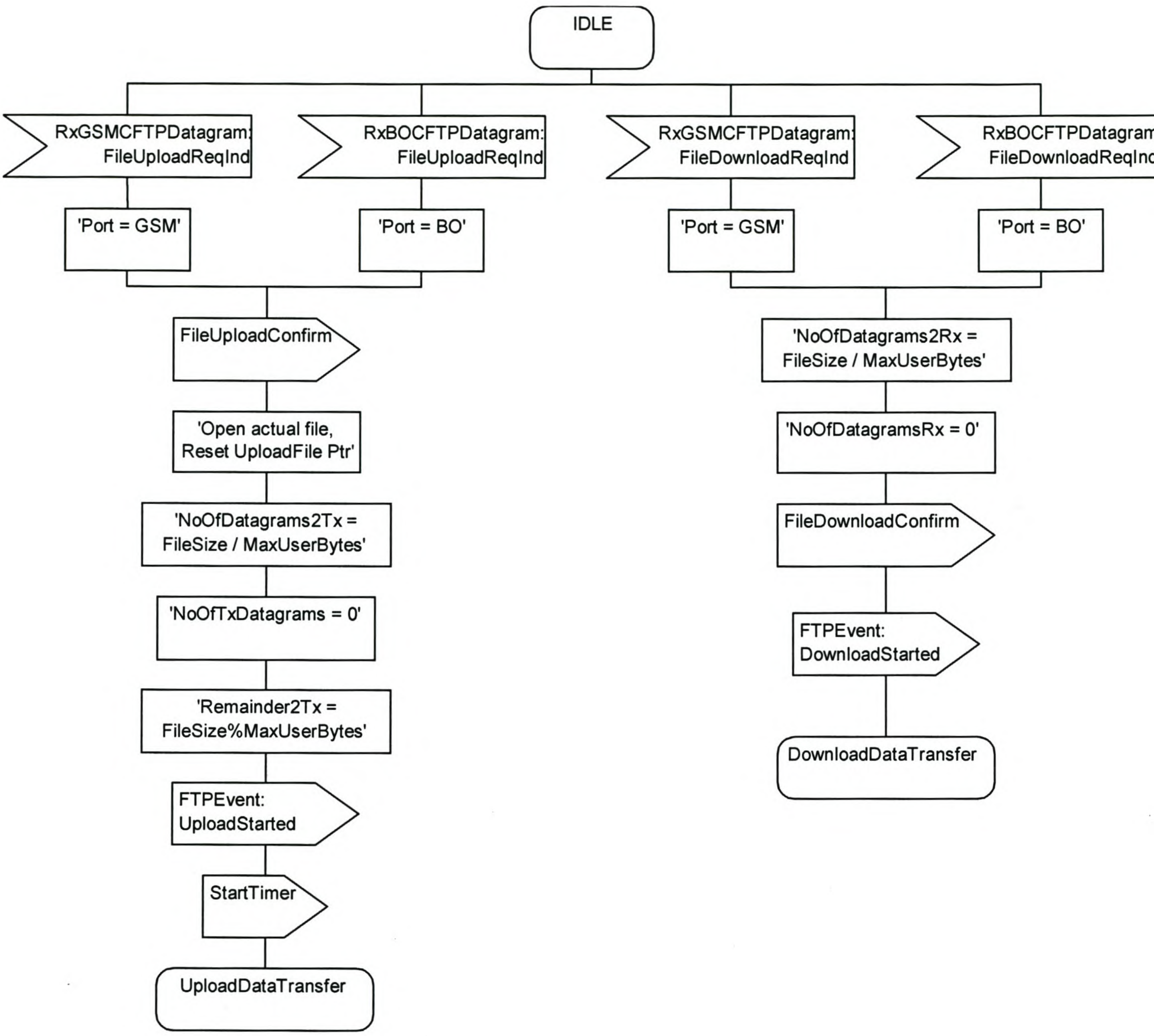

Figure A.16 FTP Manager Process

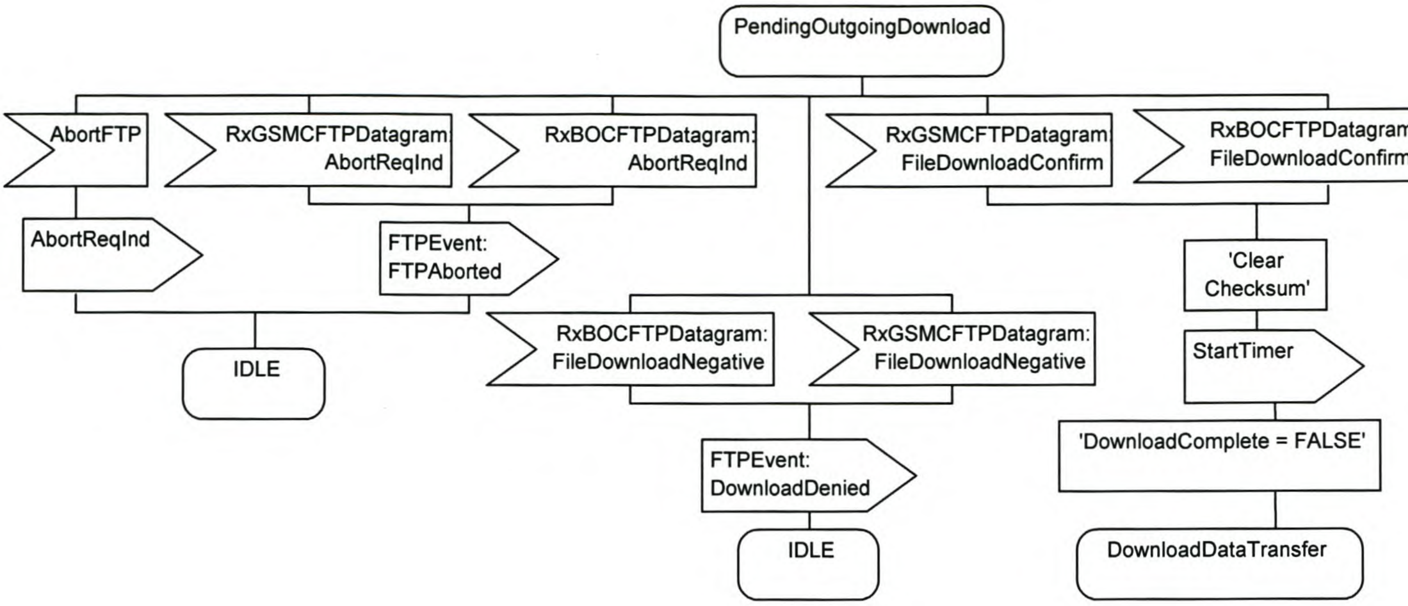

Figure A.17 FTP Manager Process

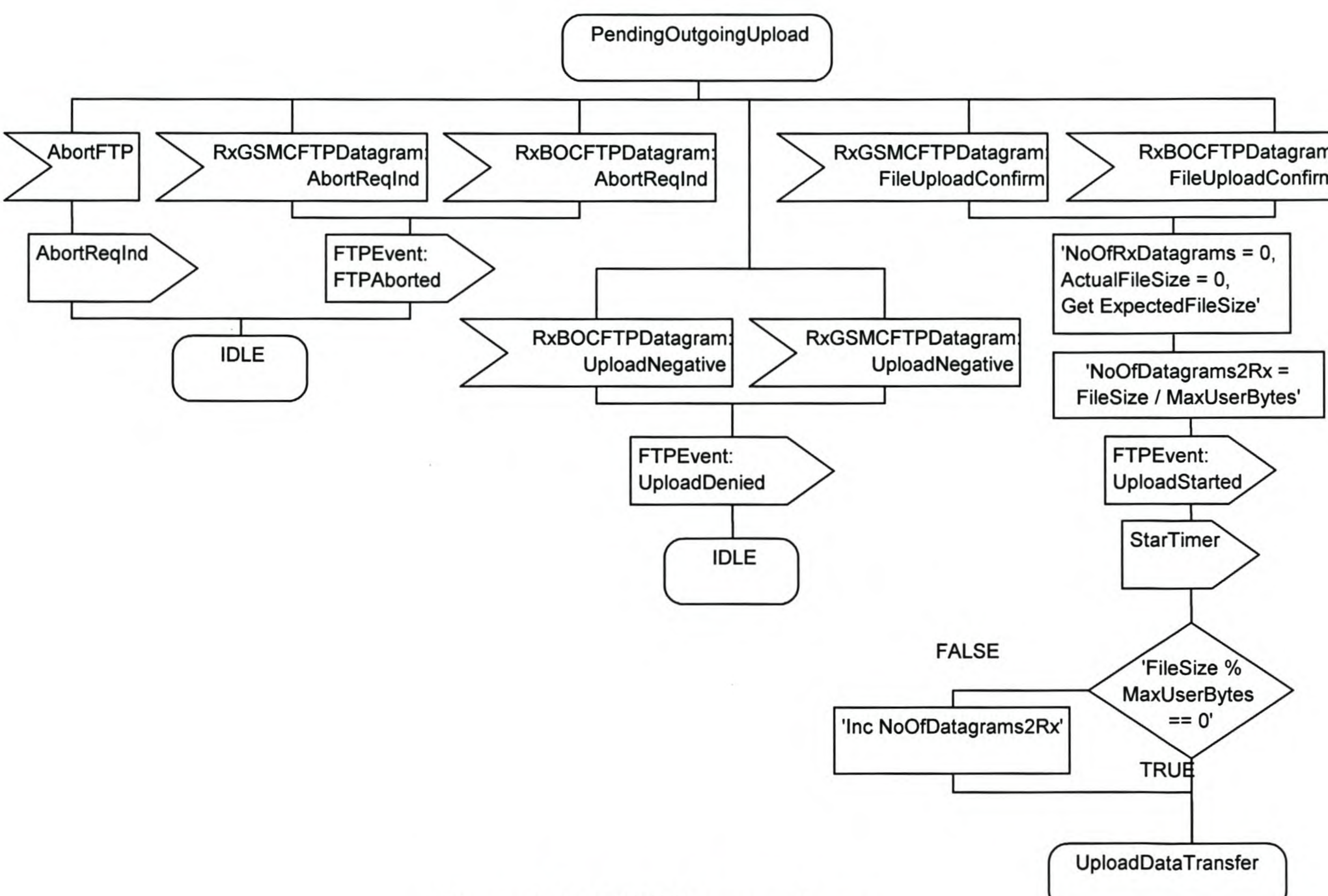

Figure A.18 FTP Manager Process

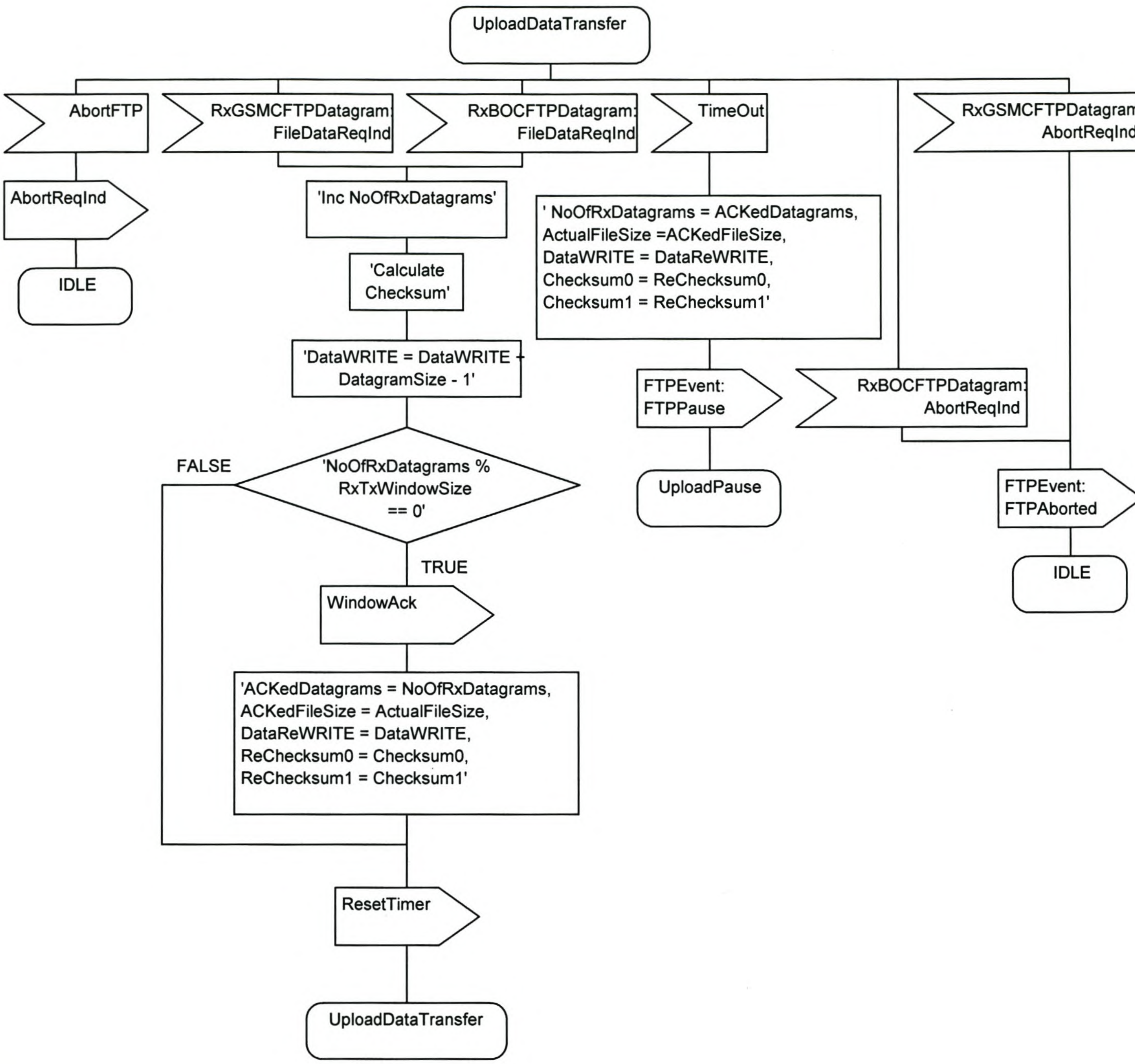

Figure A.19 FTP Manager Process

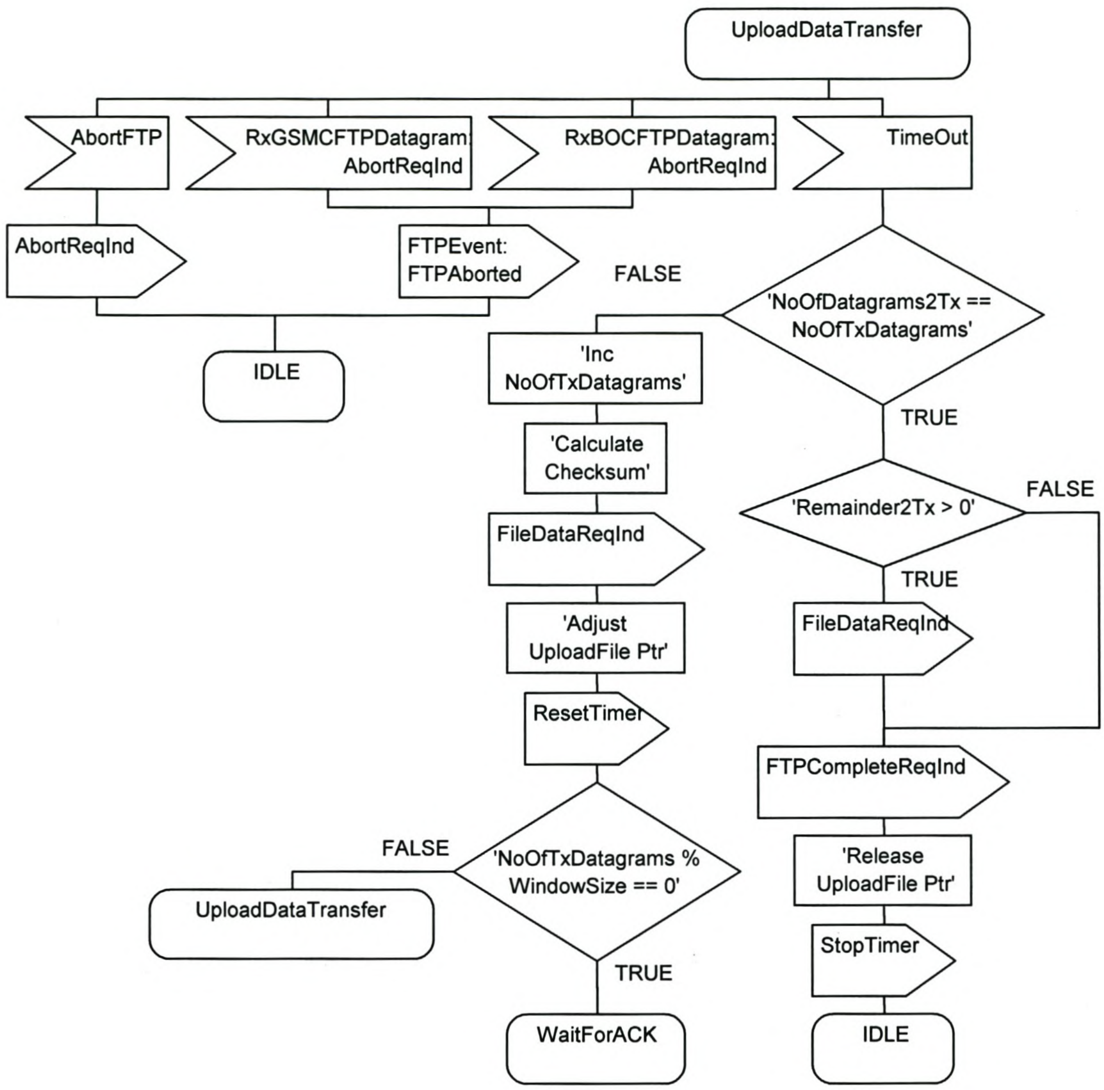

Figure A. 20 FTP Manager Process

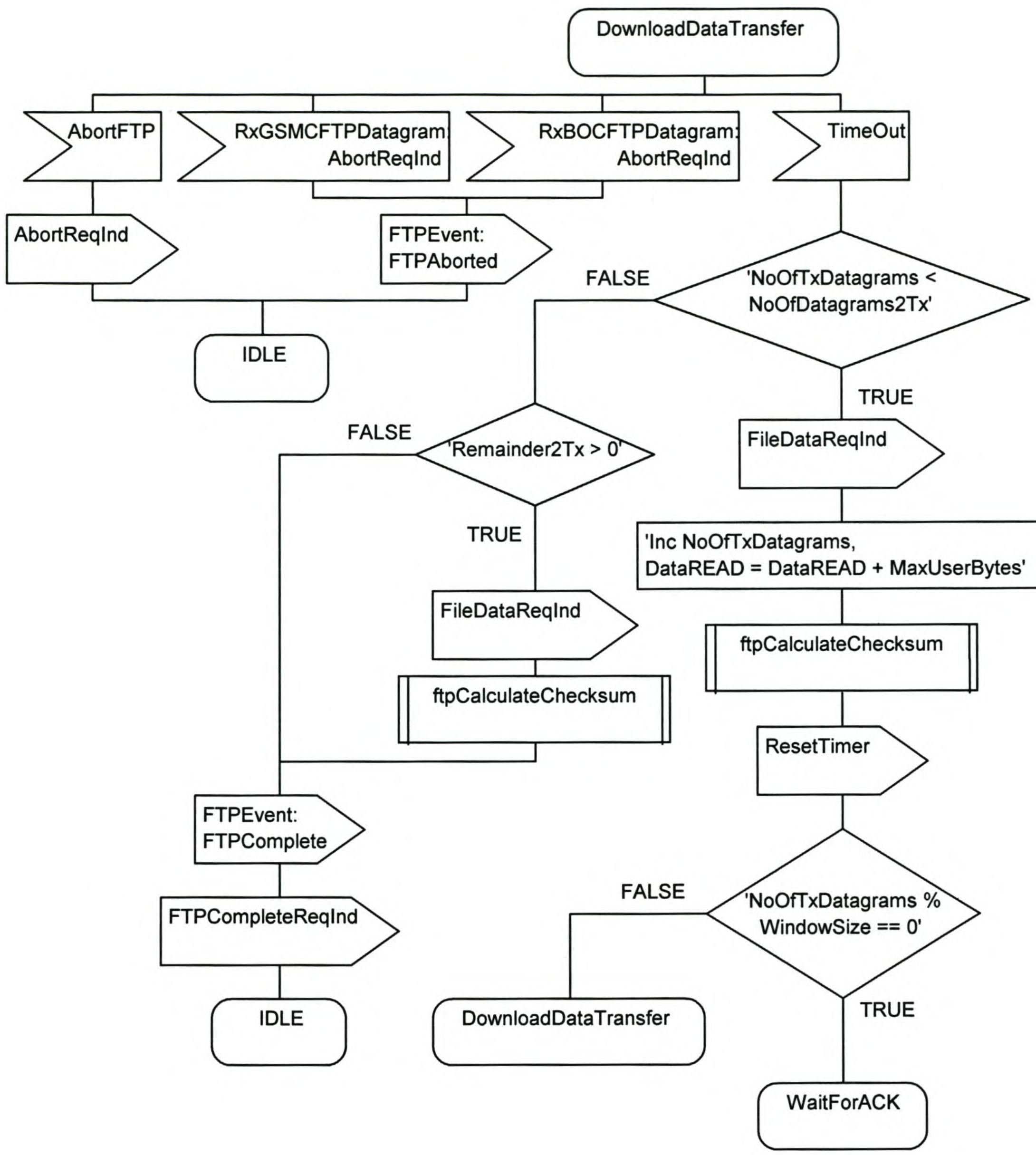

Figure A.21 FTP Manager Process

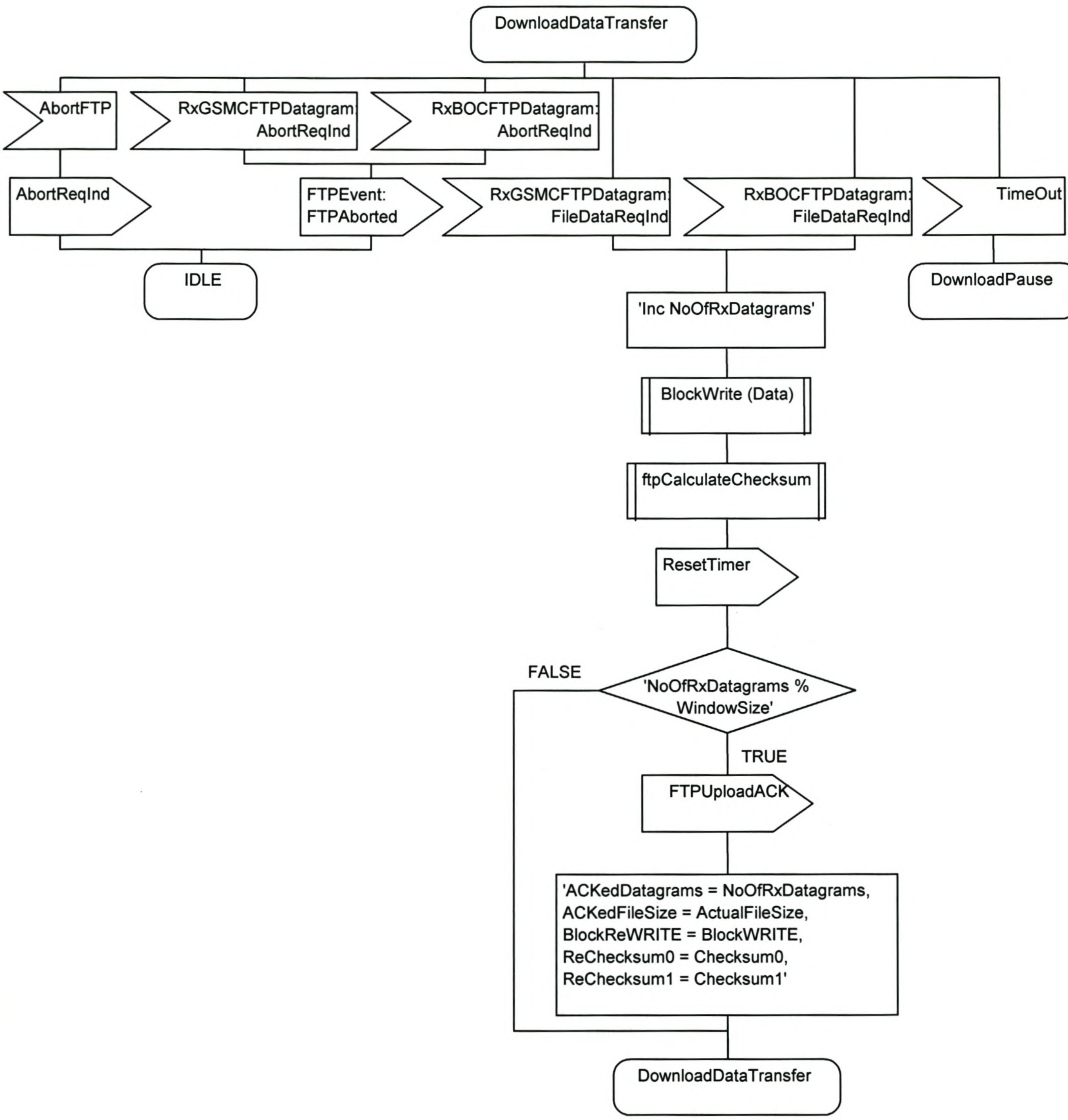

Figure A.22 FTP Manager Process

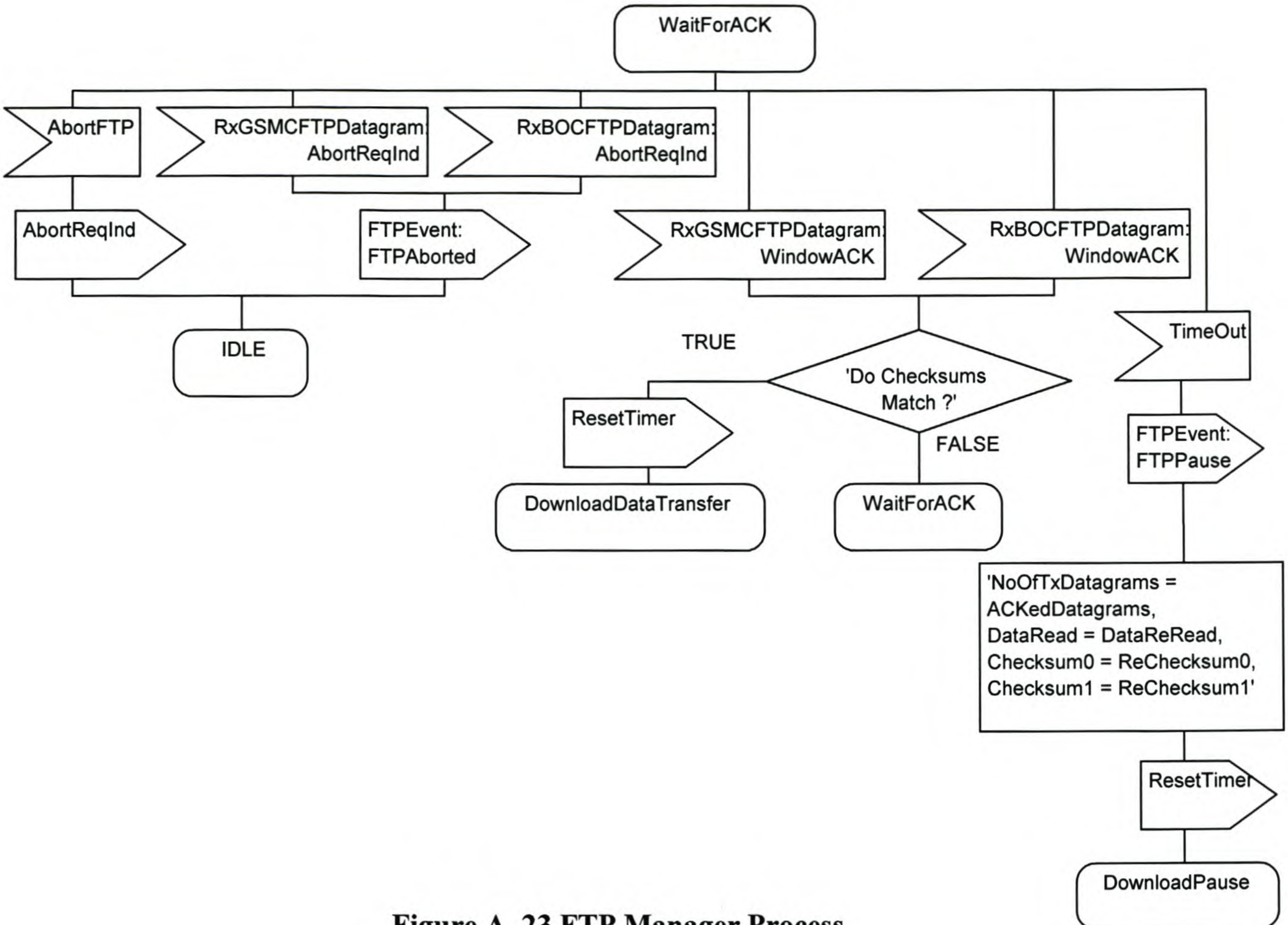

Figure A. 23 FTP Manager Process

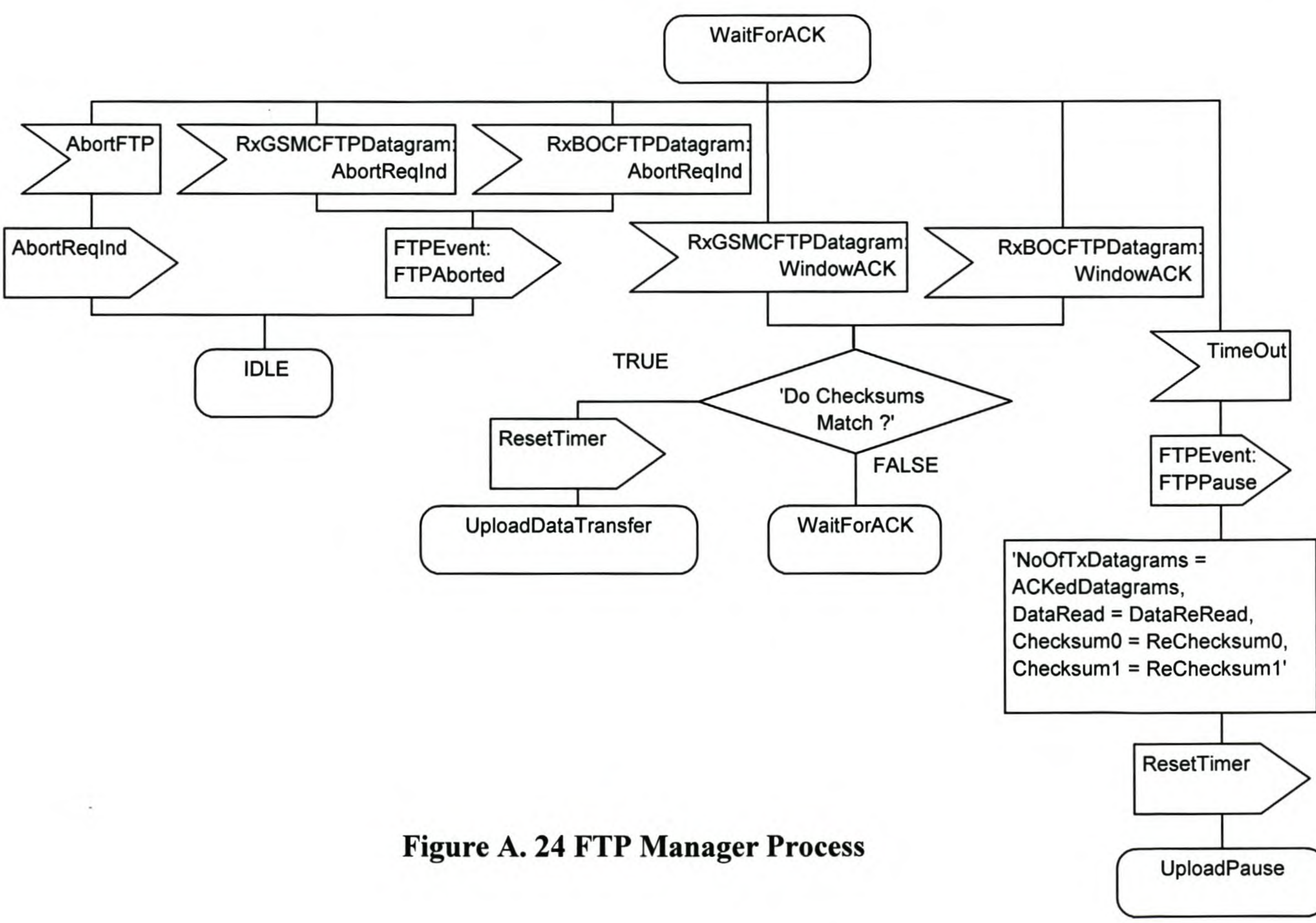

Figure A. 24 FTP Manager Process

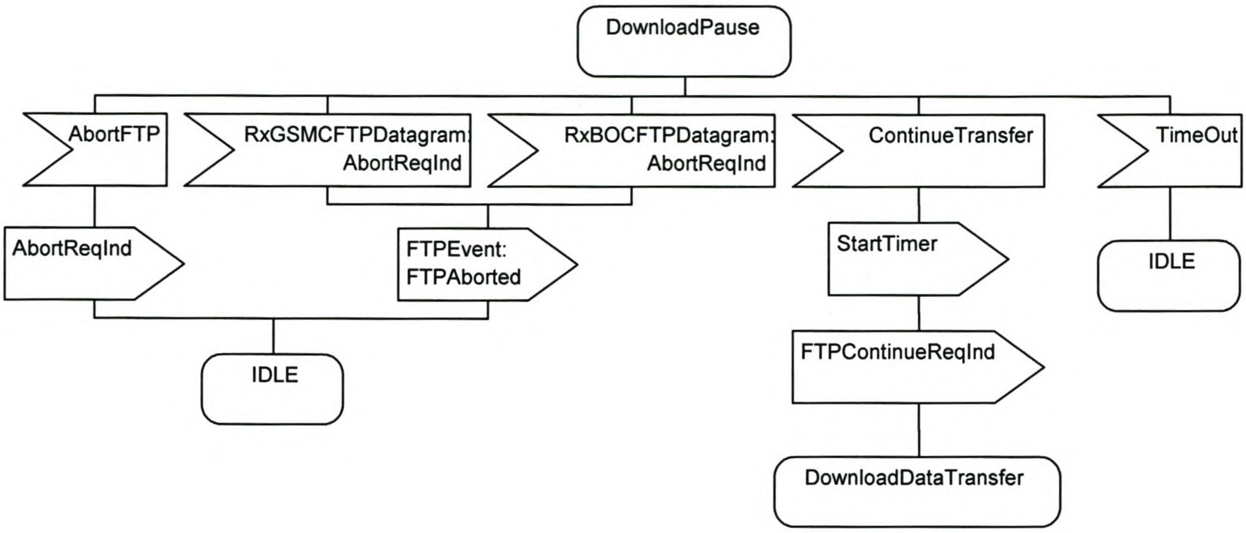

Figure A.25 FTP Manager Process

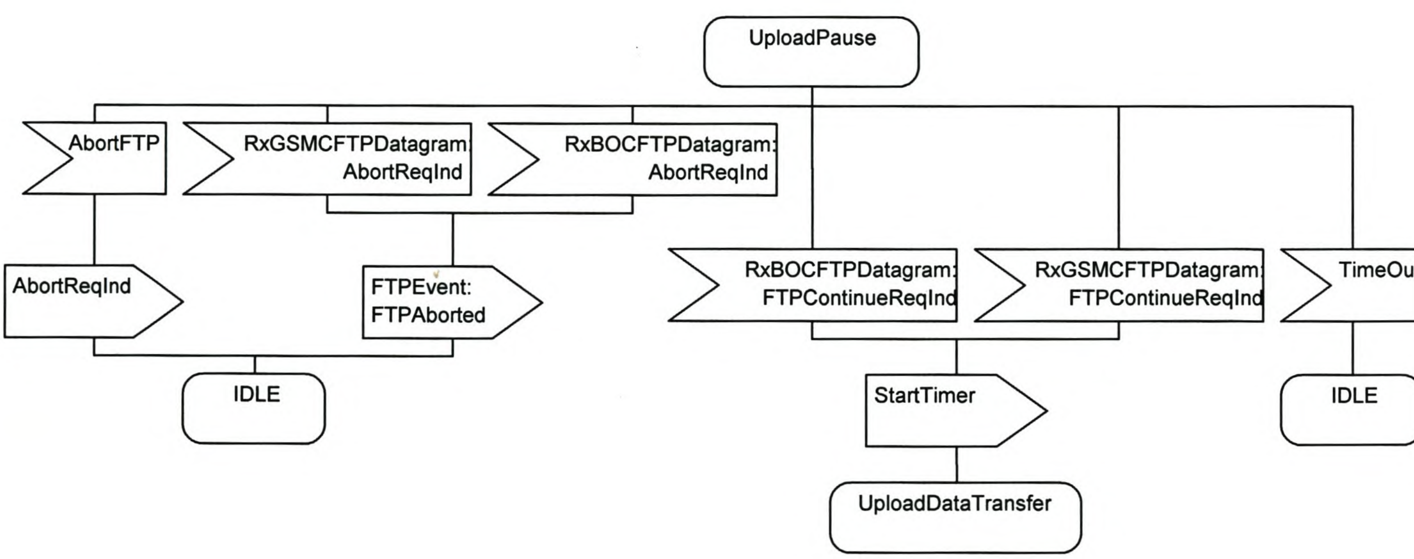

Figure A.26 FTP Manager Process

A.1.3.1 FTP Message Transmit Process

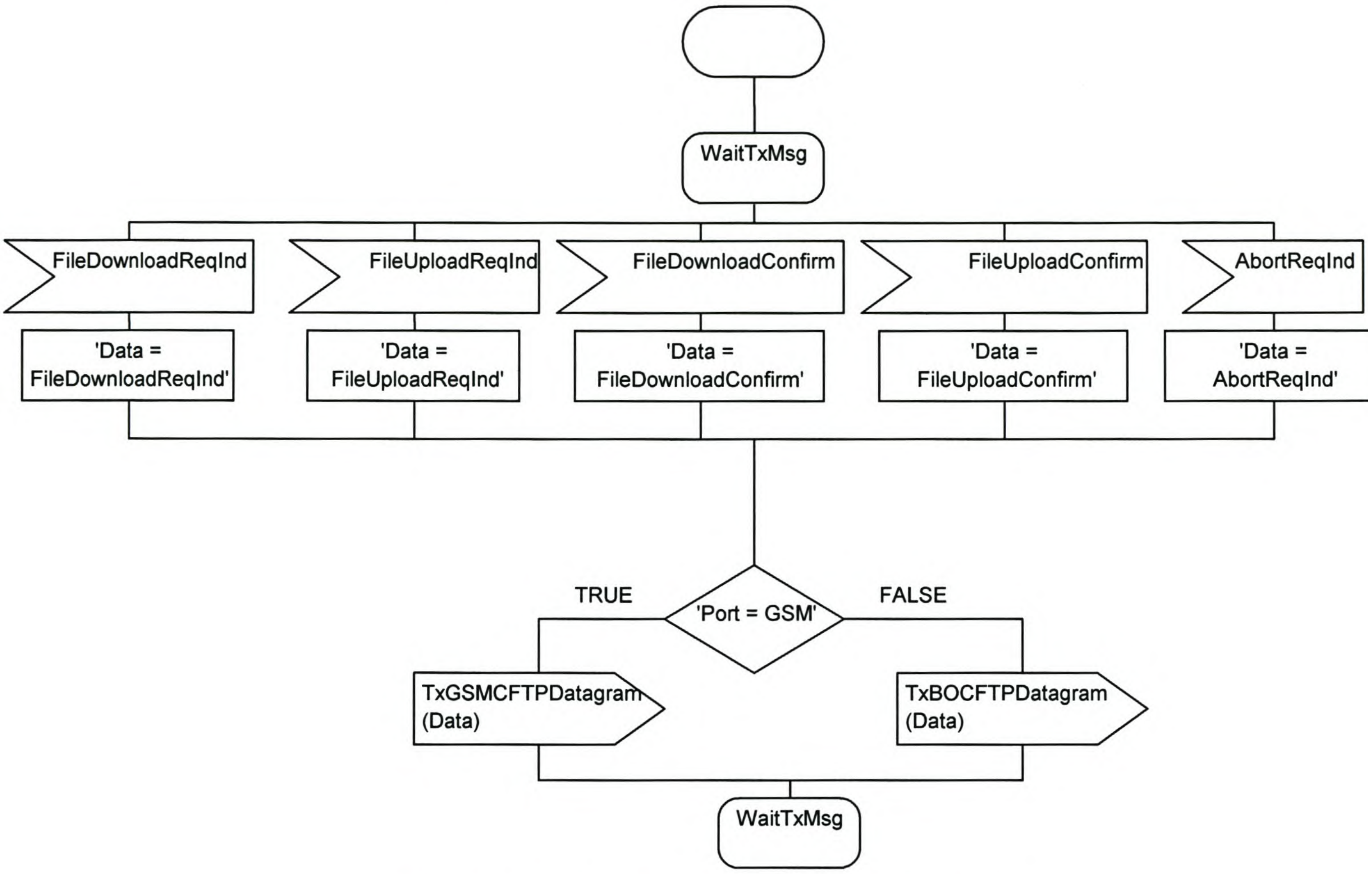

Figure A.27 FTP Message Transmit Process

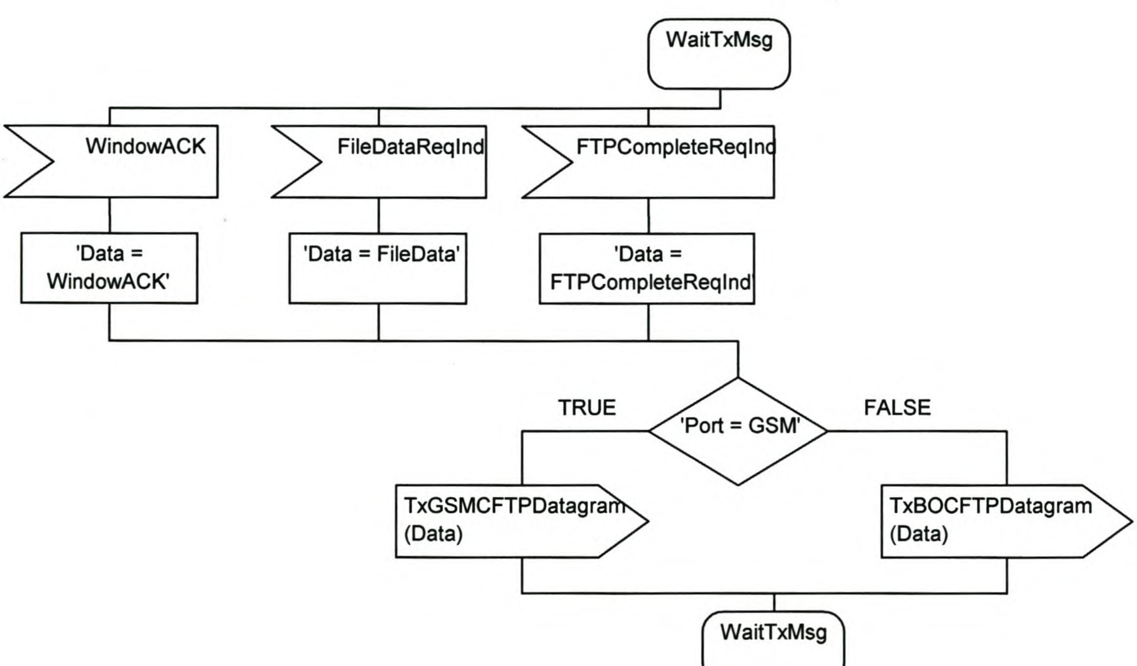

Figure A.28 FTP Message Transmit Process

A.2 WAP/AX.25 System Diagram

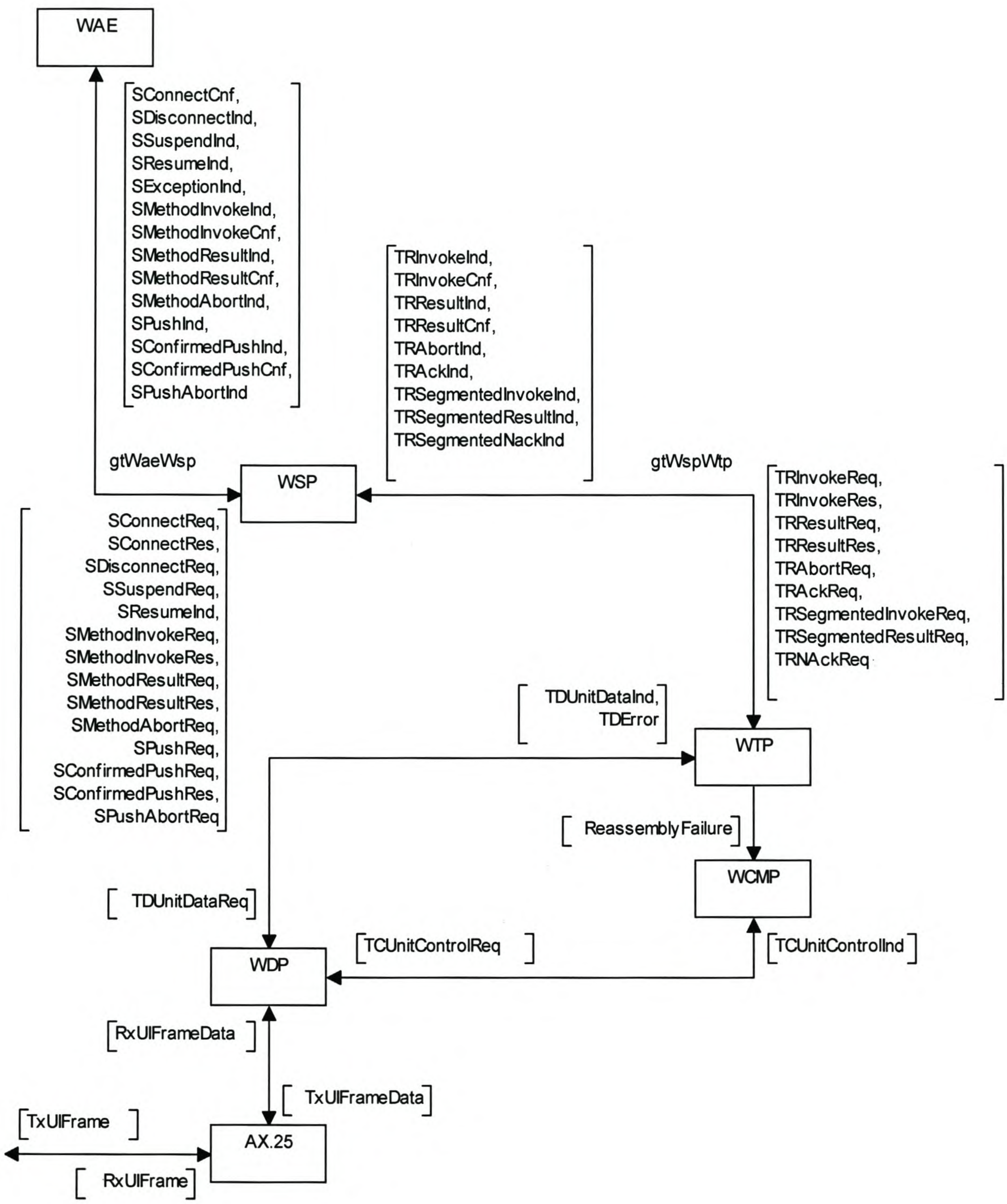

Figure A.29 WAP/AX.25 System Diagram

A.2.1 WTP Protocol Layer (Client)

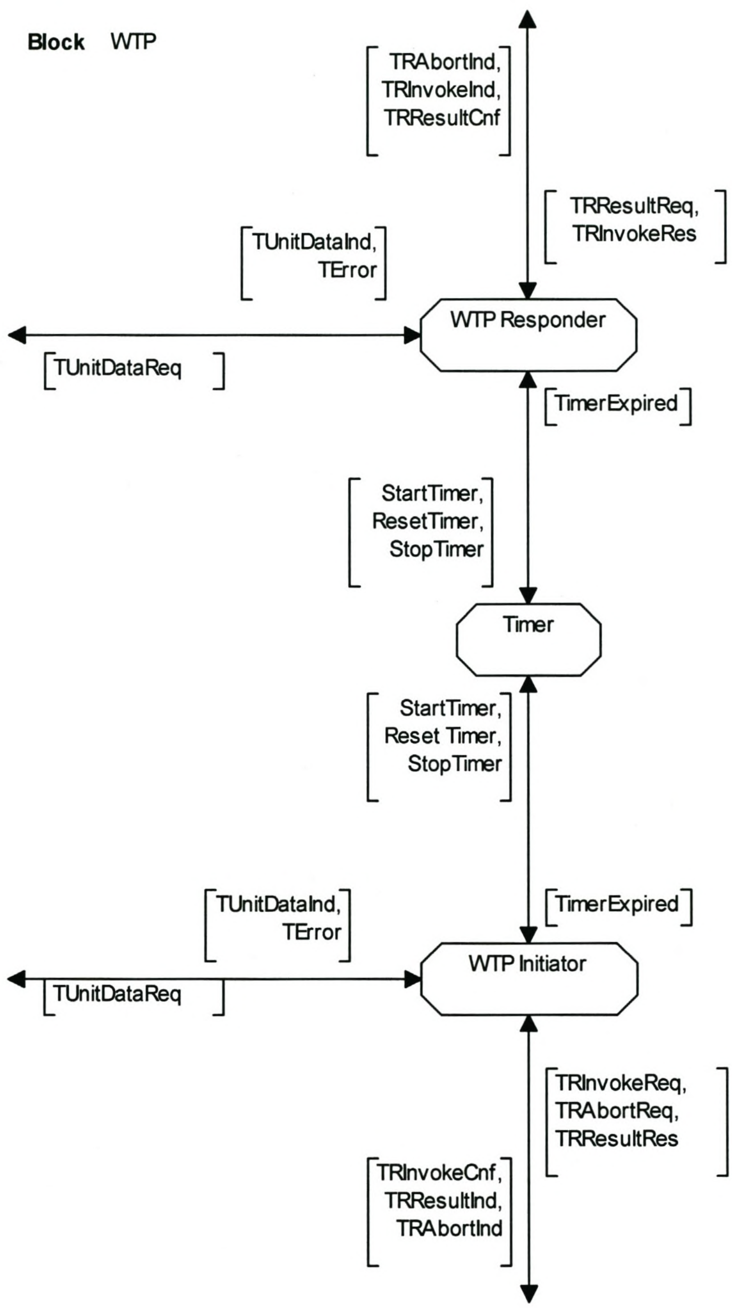

Figure A.30 WTP Initiator and Responder

A.2.1.1 WTP Initiator Process

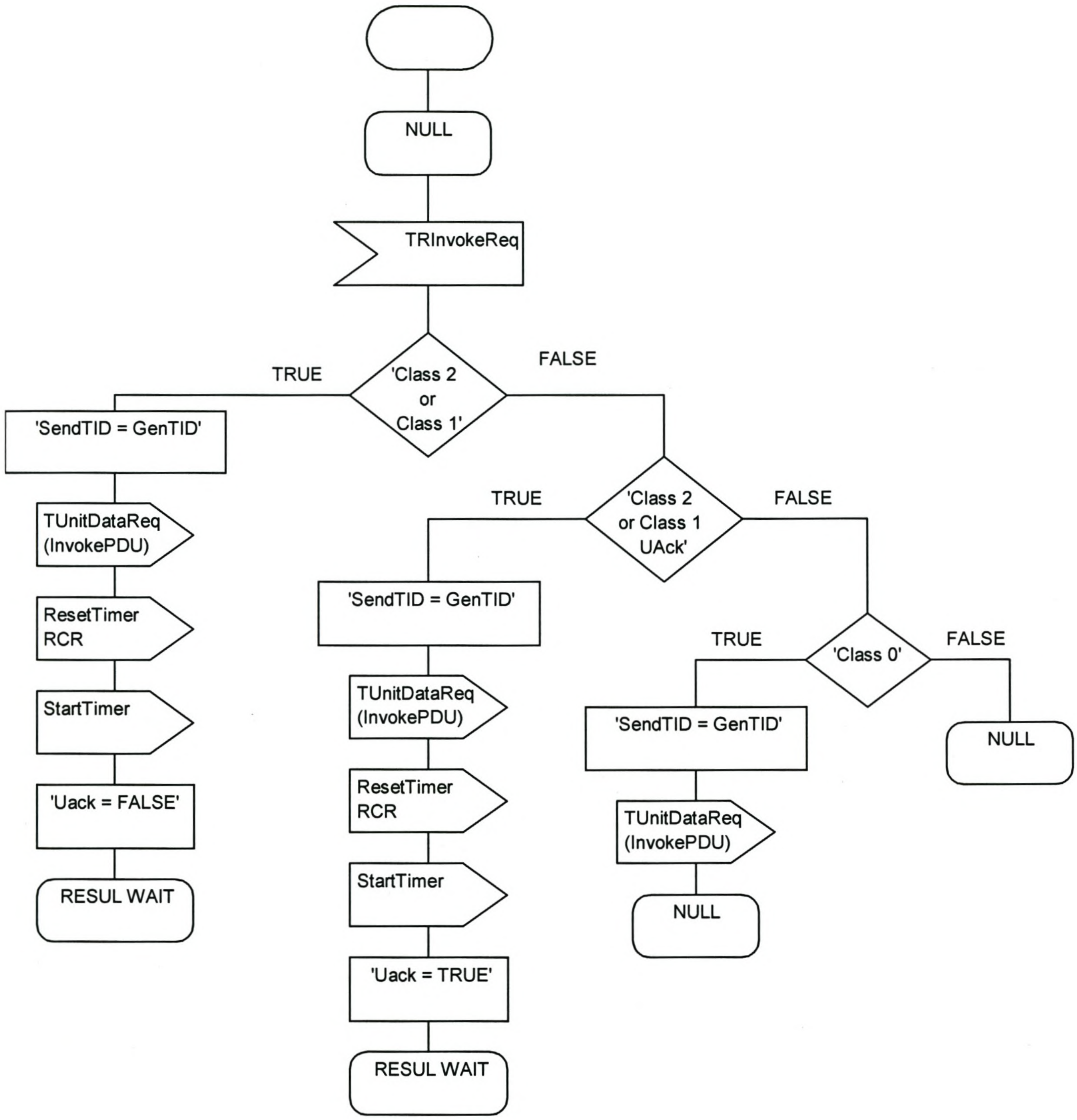

Figure A.31 WTP Initiator Process

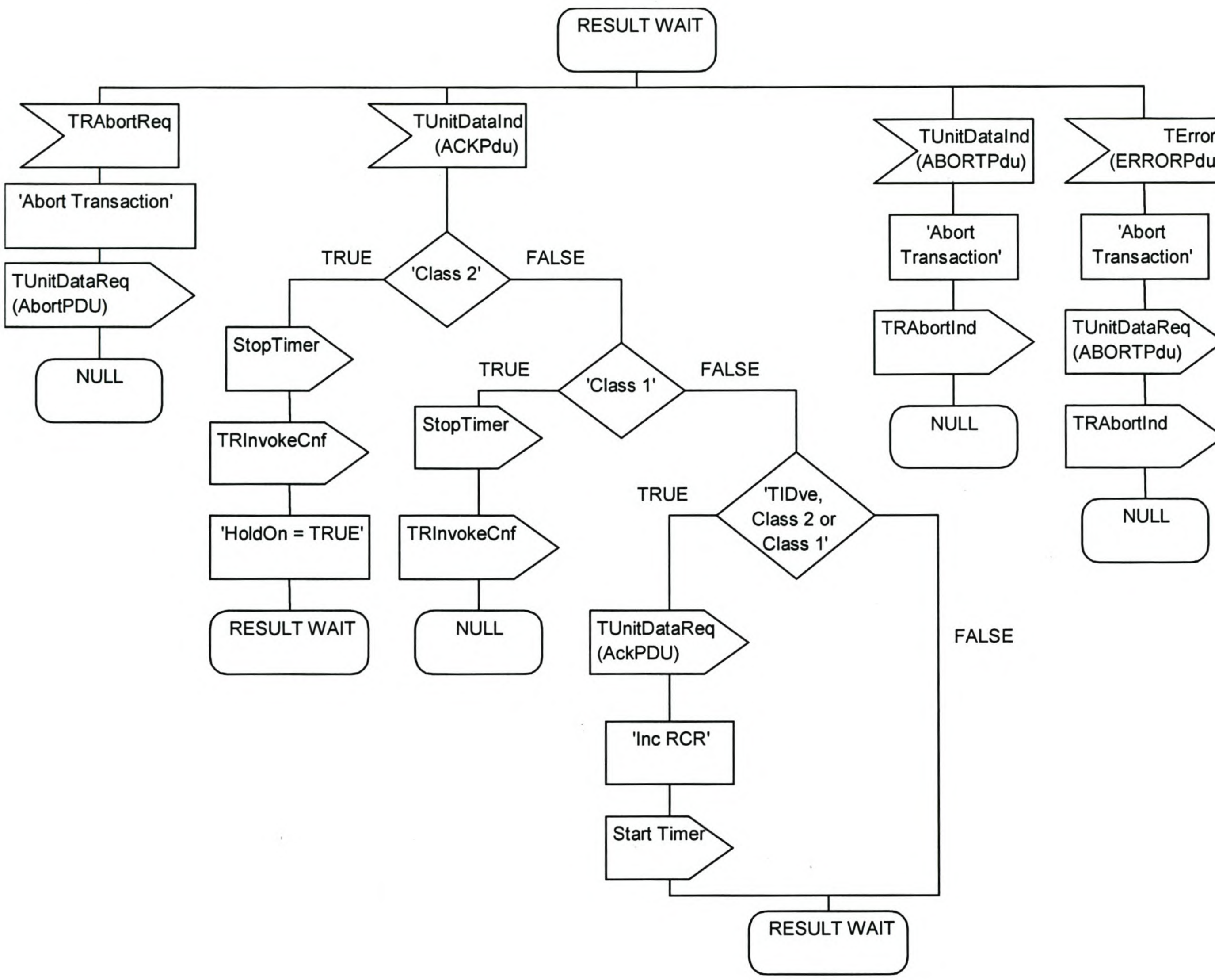

Figure A.32 WTP Initiator Process

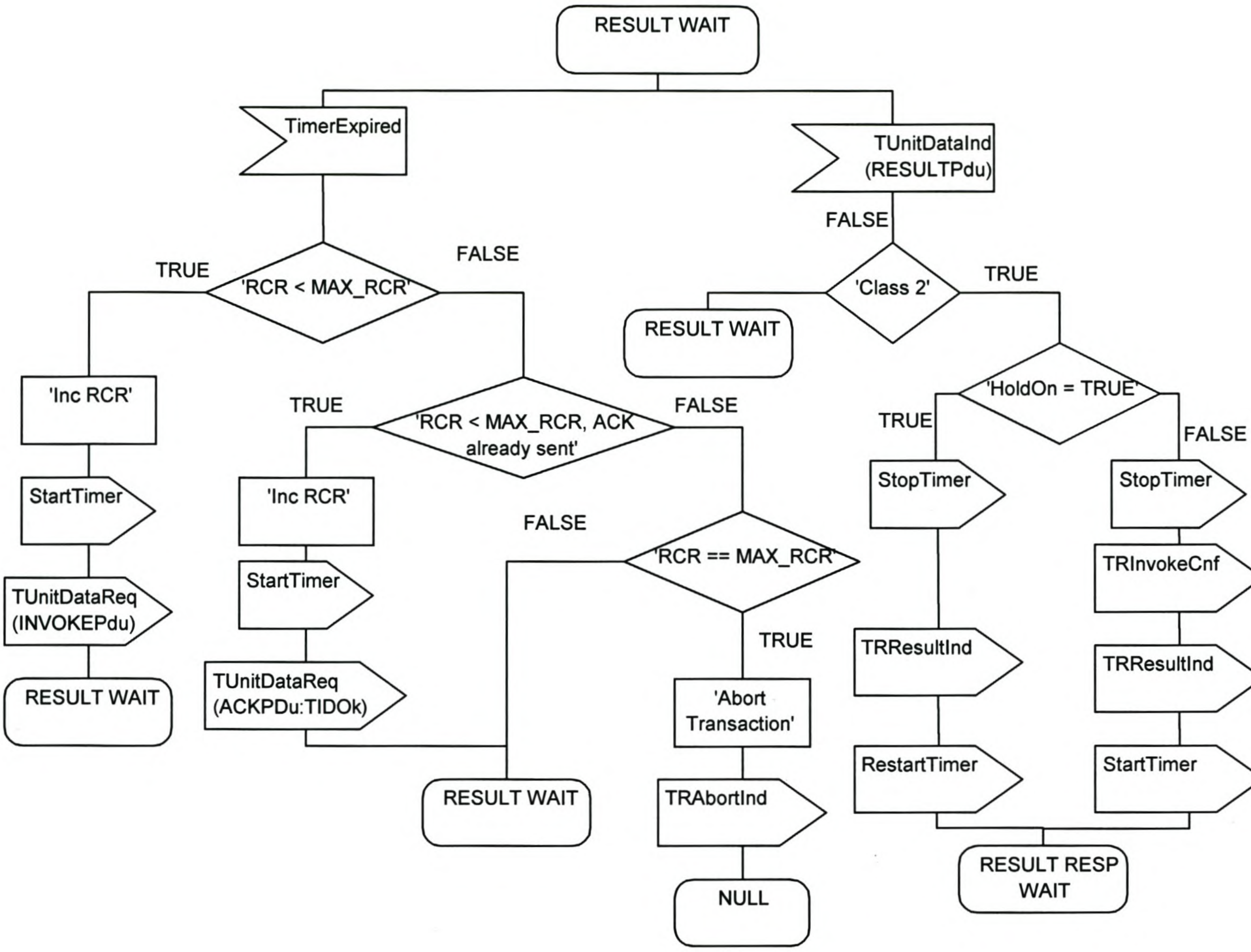

Figure A.33 WTP Initiator Process

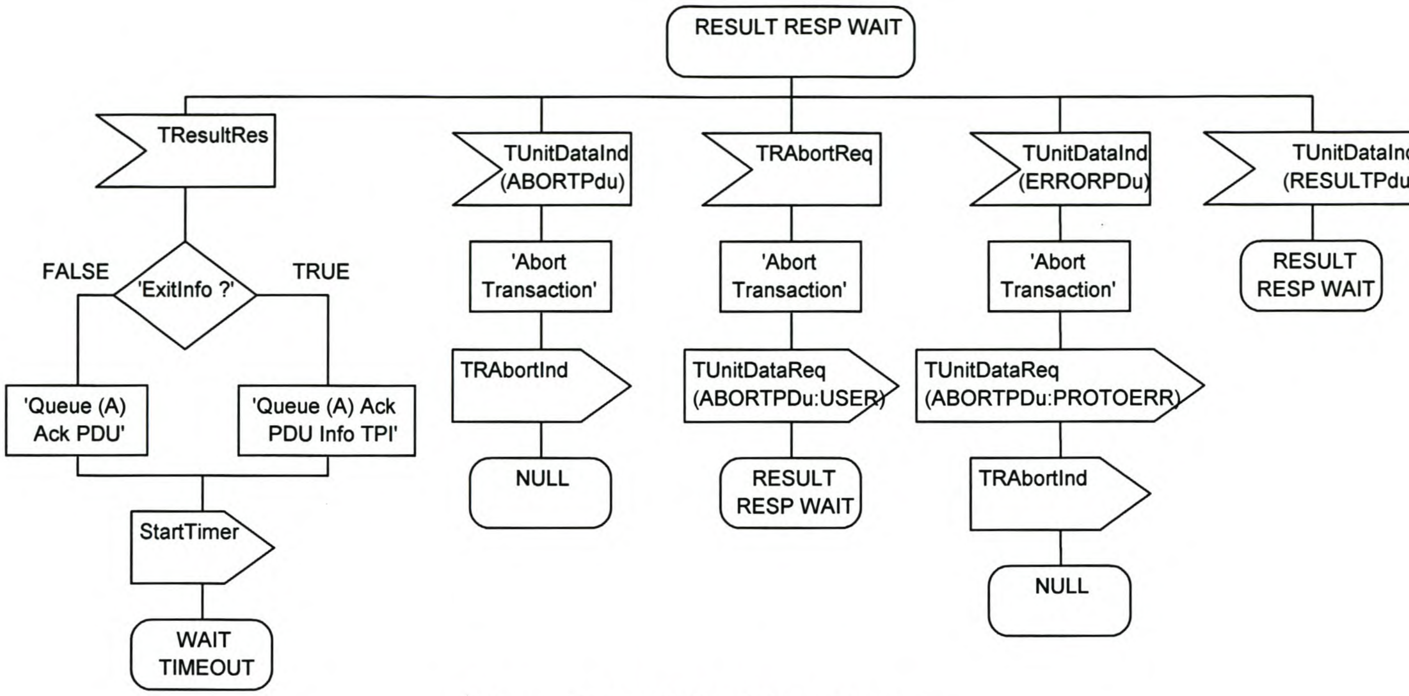

Figure A.34 WTP Initiator Process

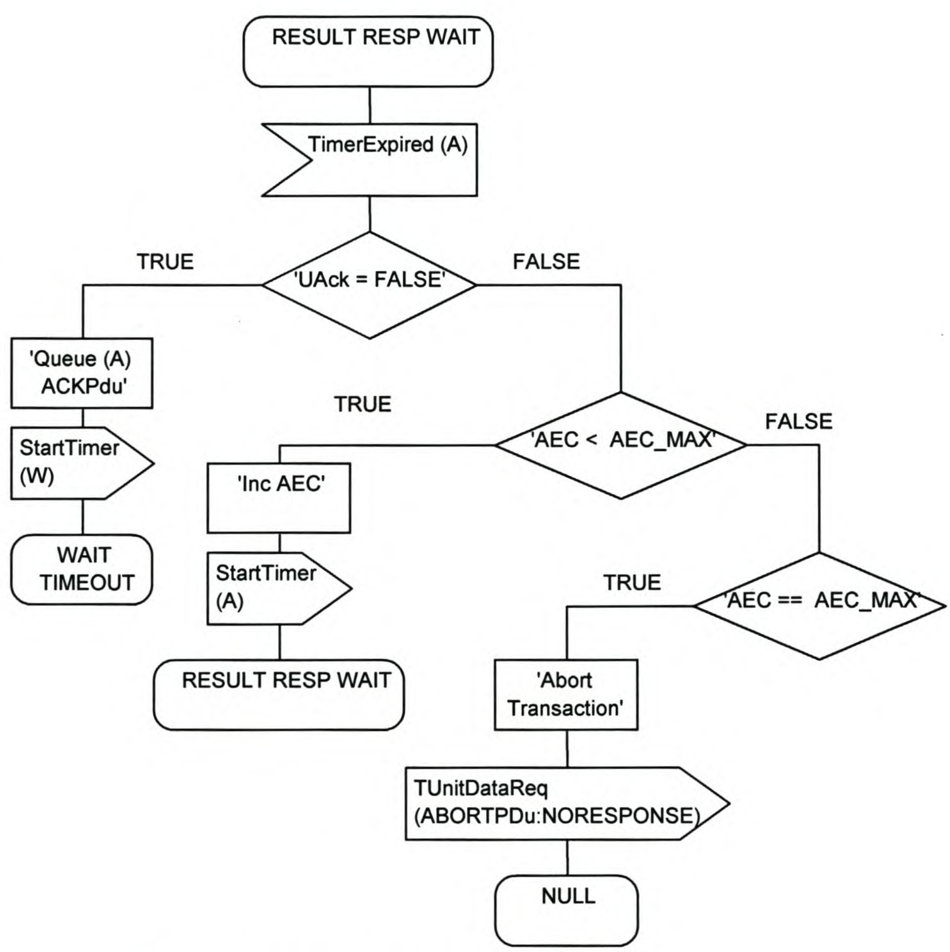

Figure A. 35 WTP Initiator Process

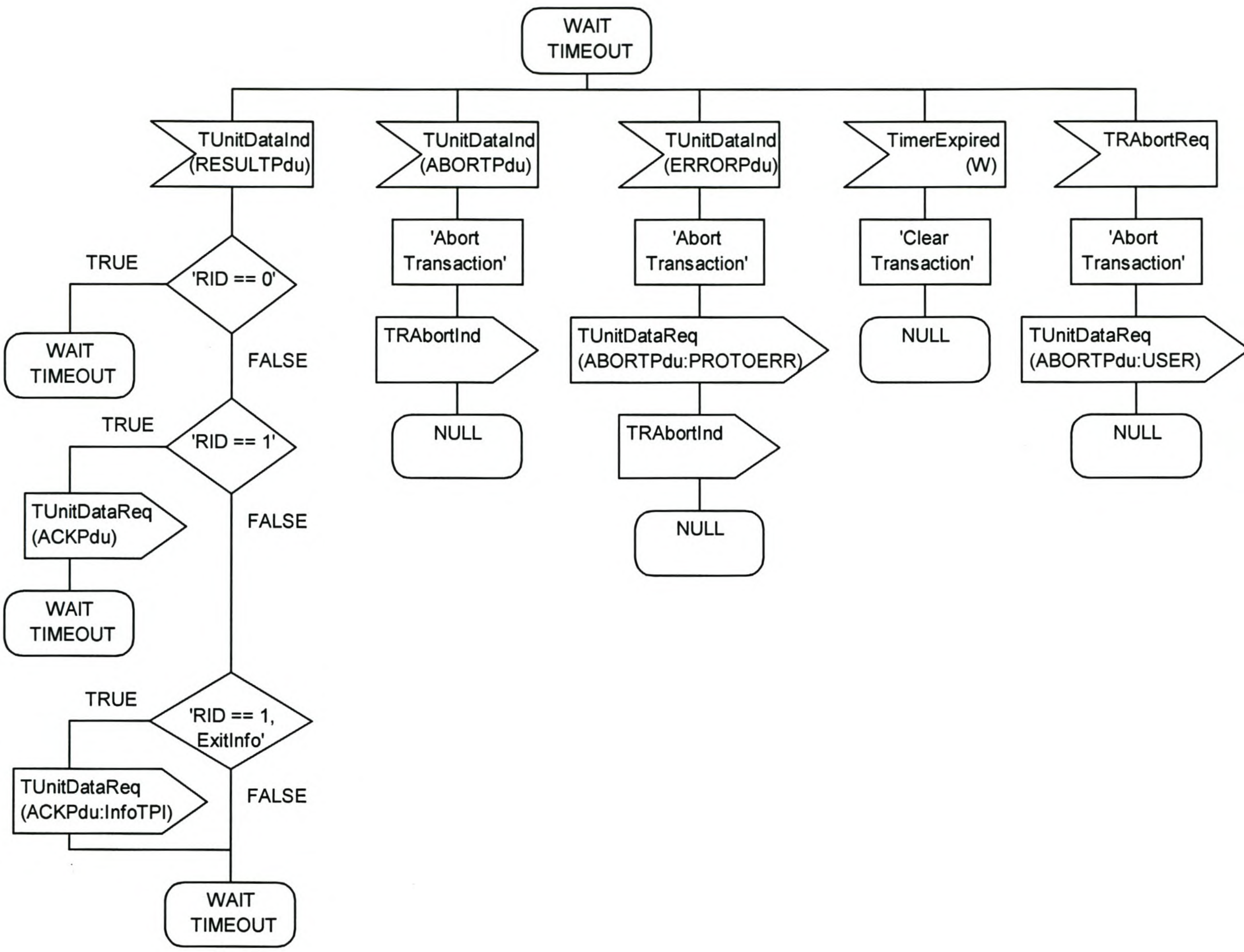

Figure A.36 WTP Initiator Process

A.2.1.2 WTP Responder (Client)

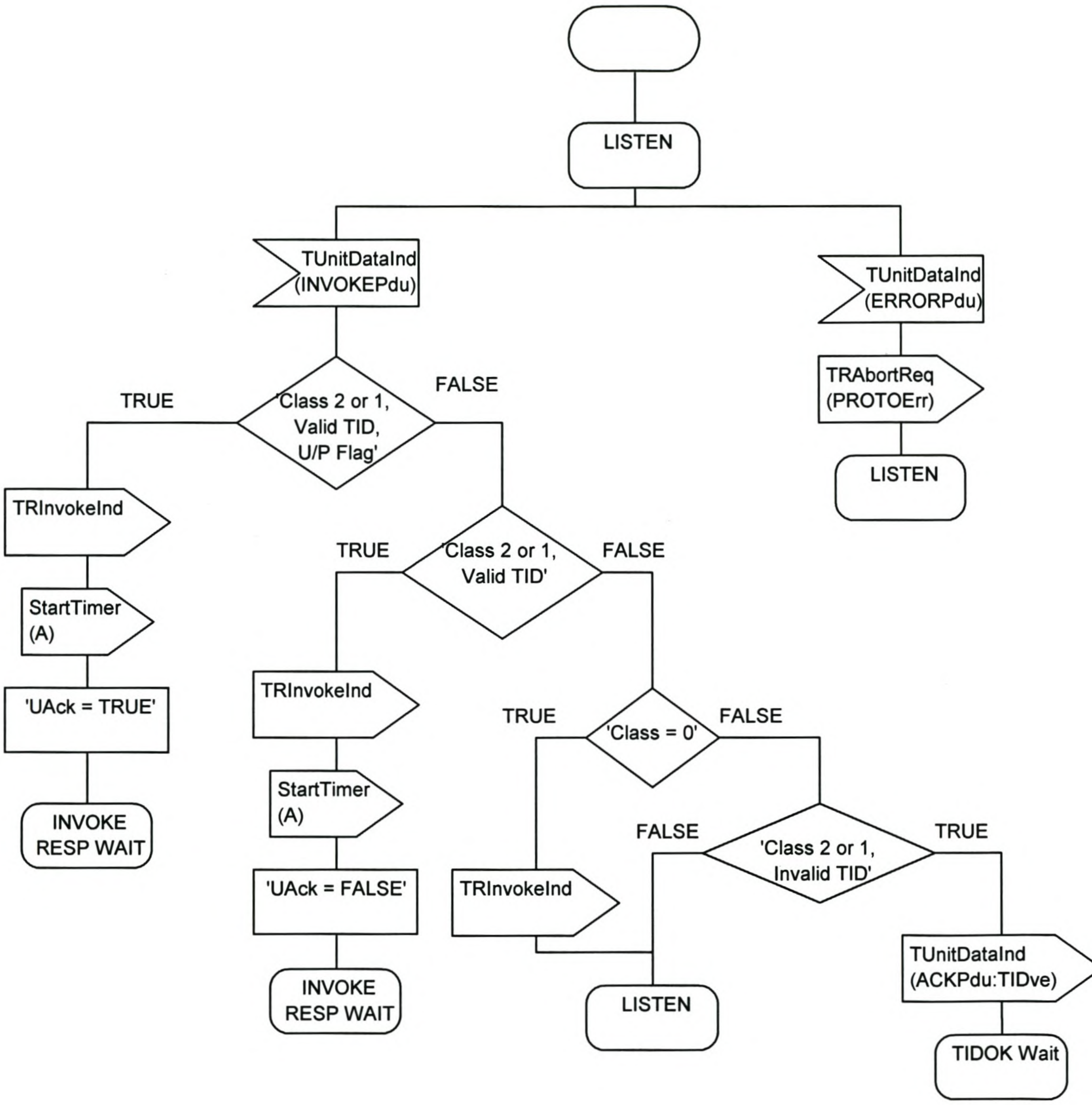

Figure A.37 Client WTP Responder Process

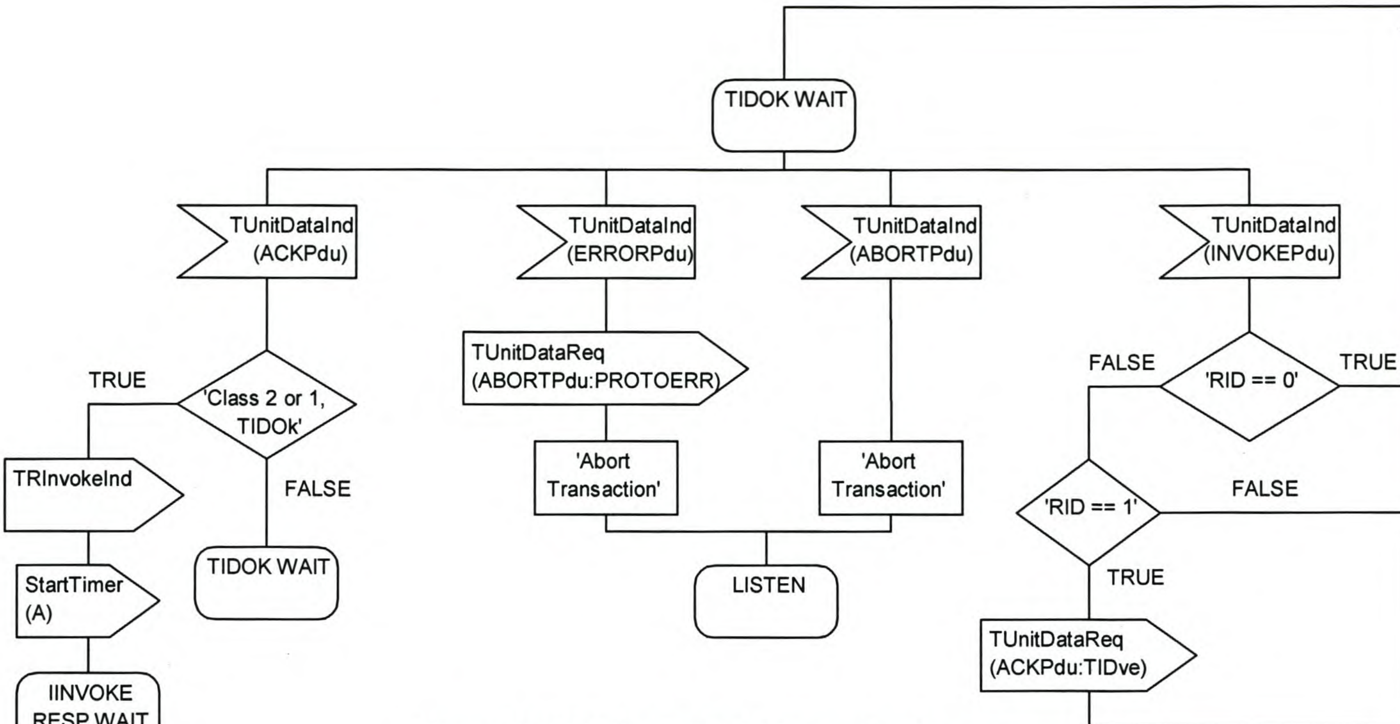

Figure A. 38 Client WTP Responder Process

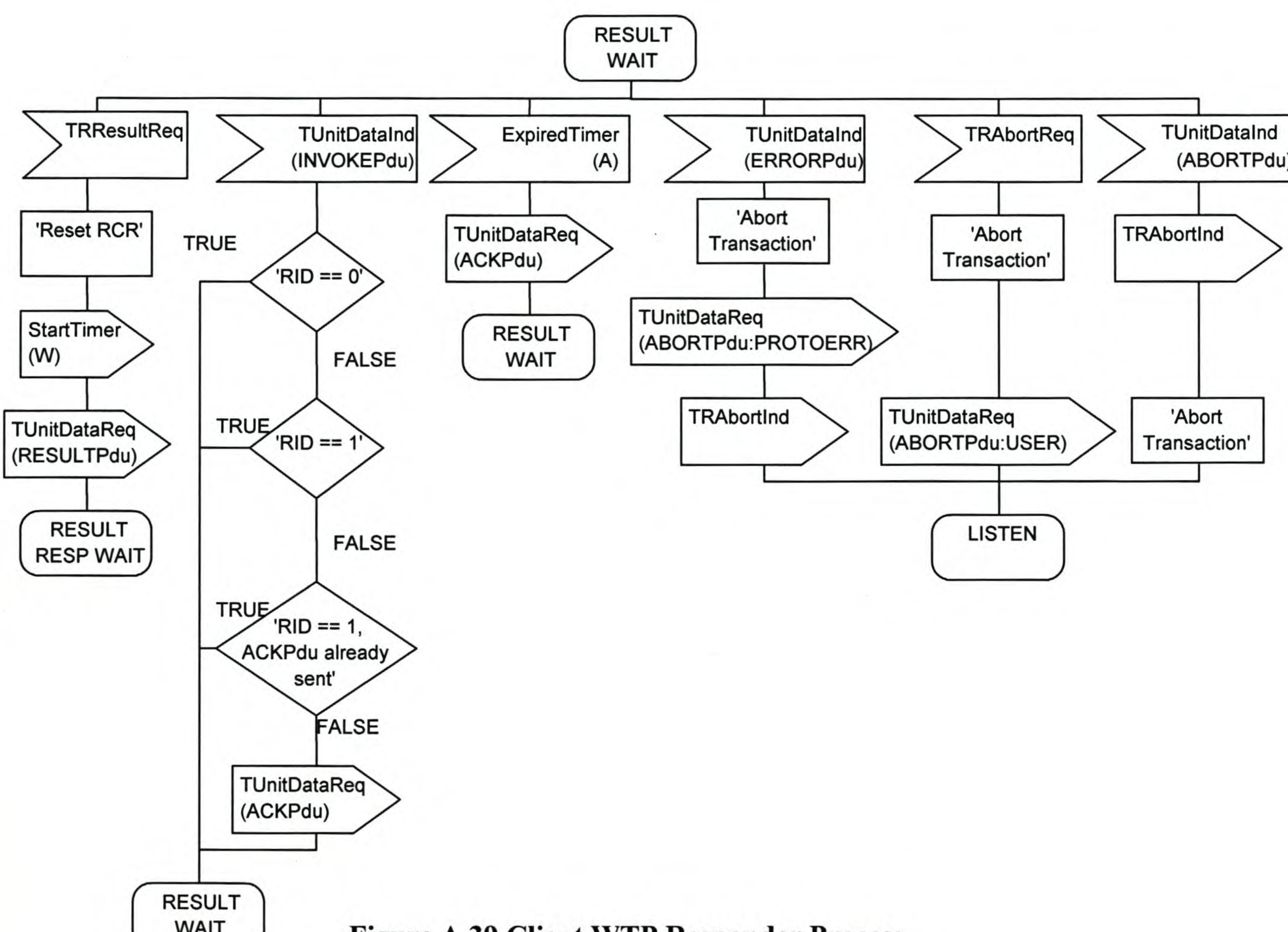

Figure A.39 Client WTP Responder Process

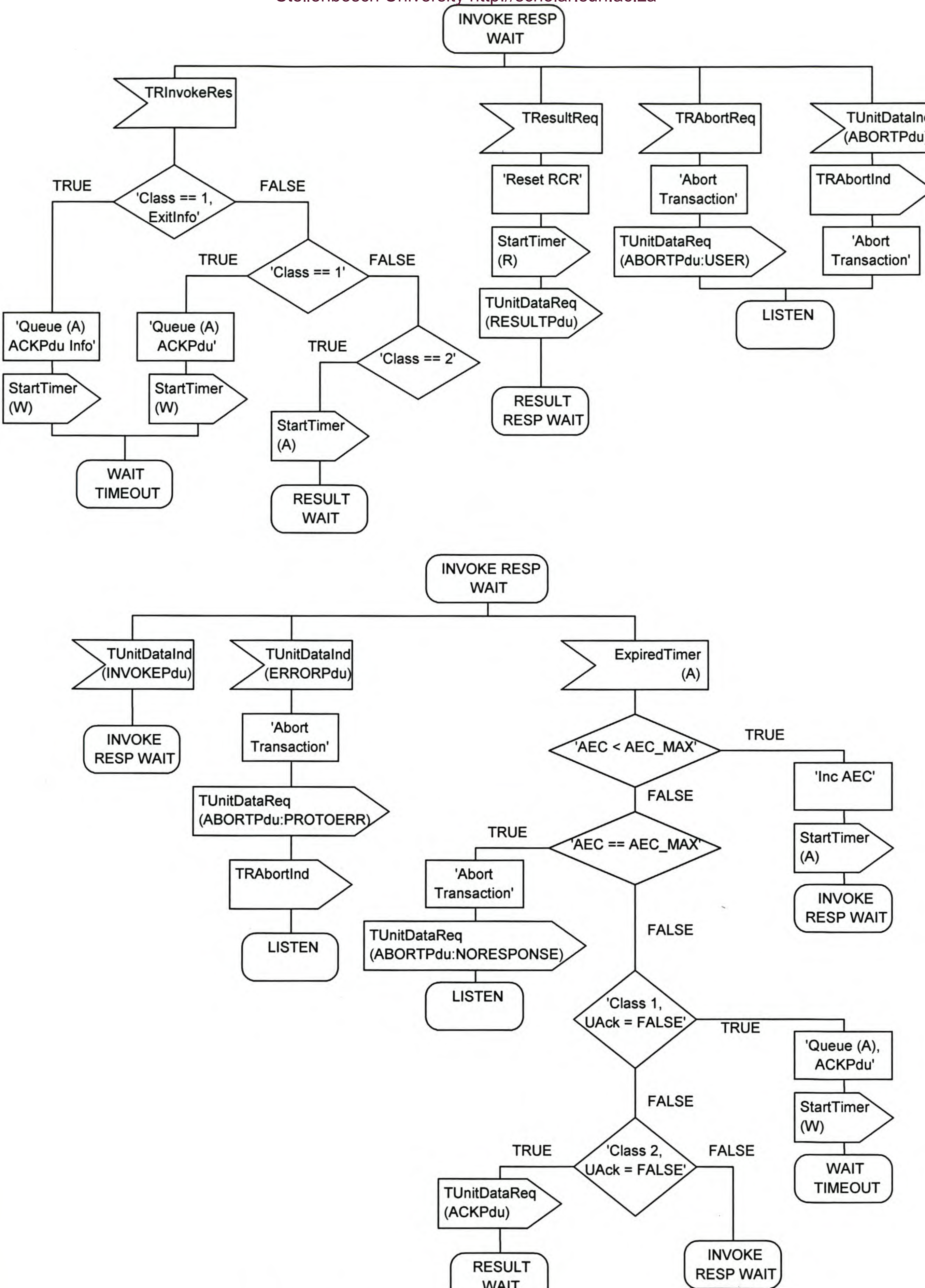

Figure A.40 Client WTP Responder Process

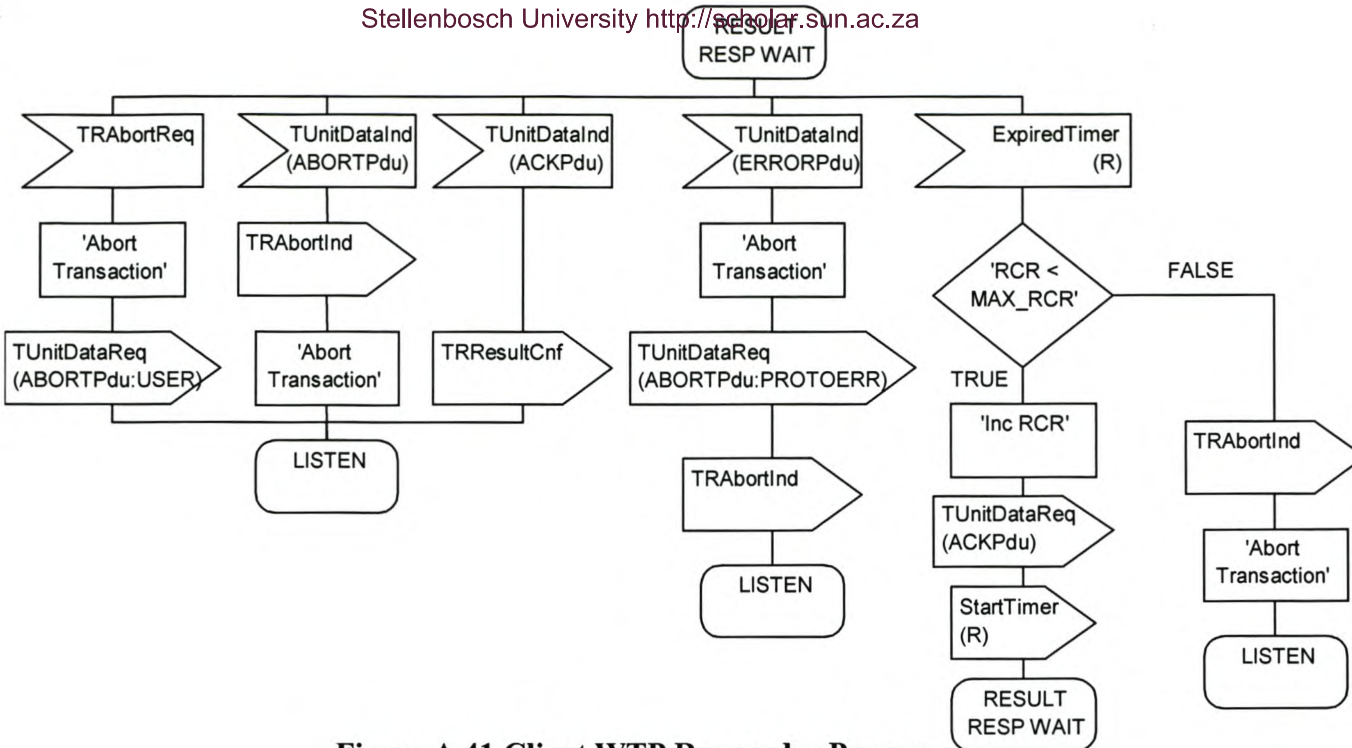

Figure A.41 Client WTP Responder Process

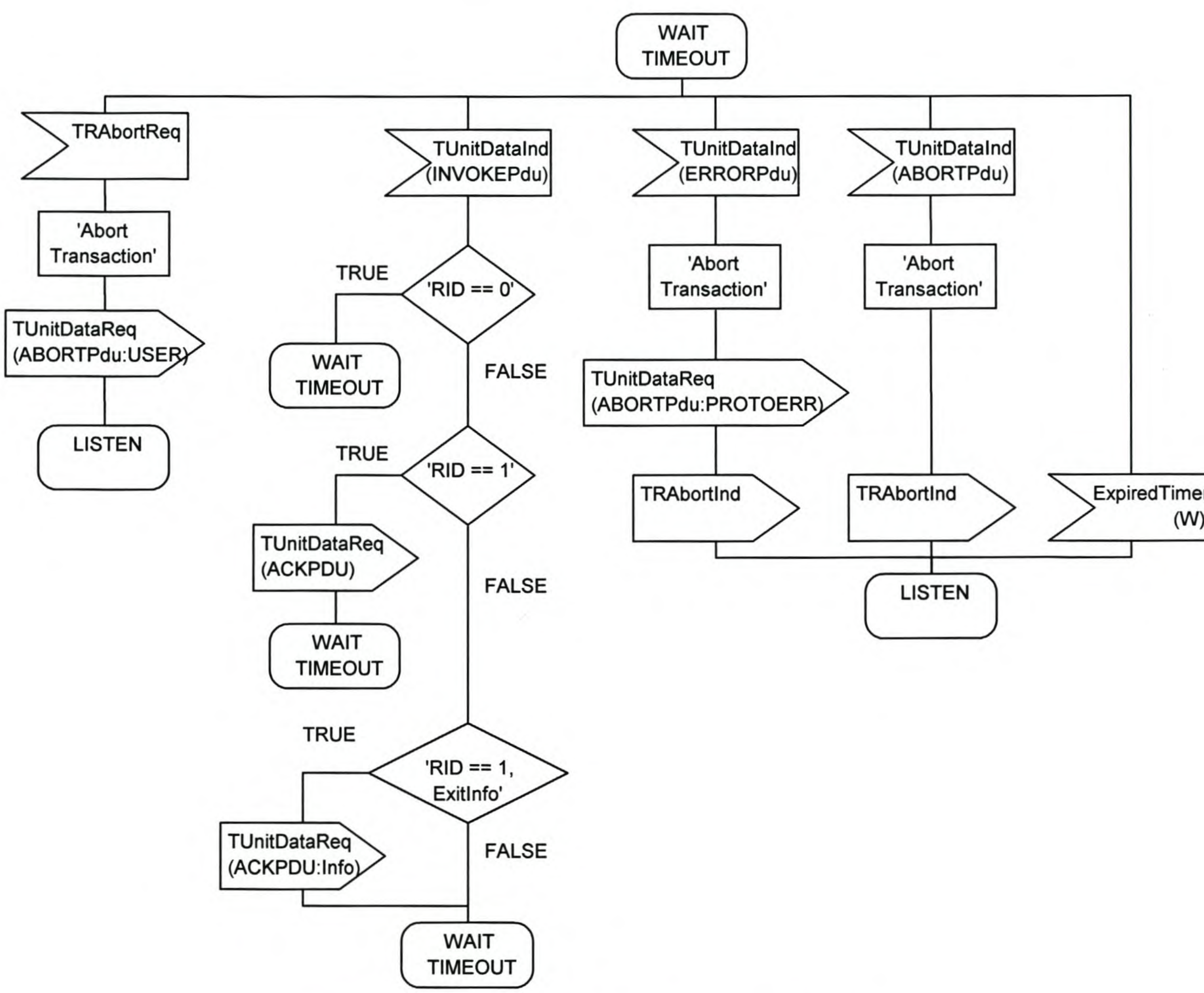

Figure A.42 Client WTP Responder Process

A.2.2 WSP Protocol Layer (Client)

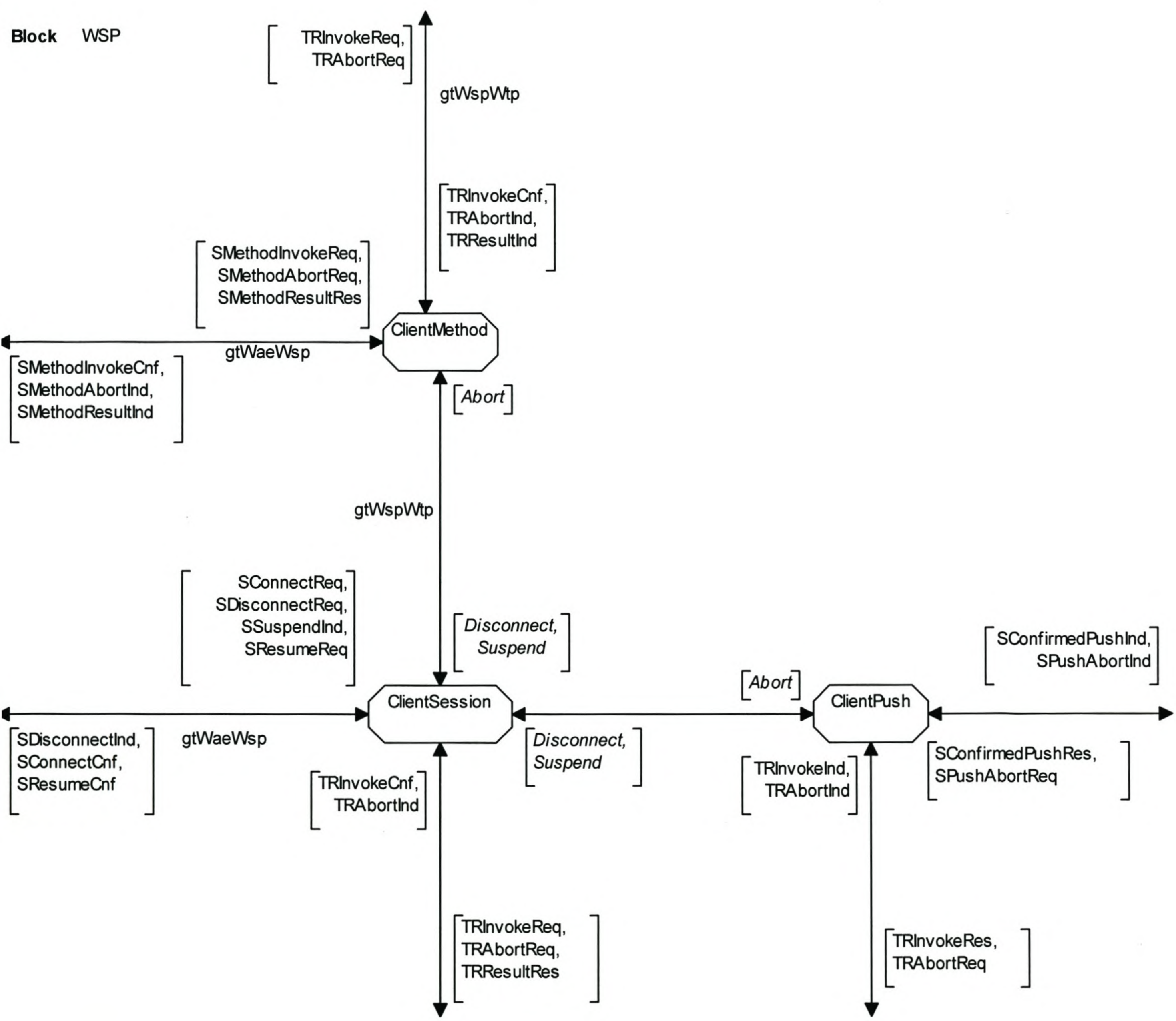

Figure A.43 Client WSP Processes

A.2.2.1 Client Session Process (Client)

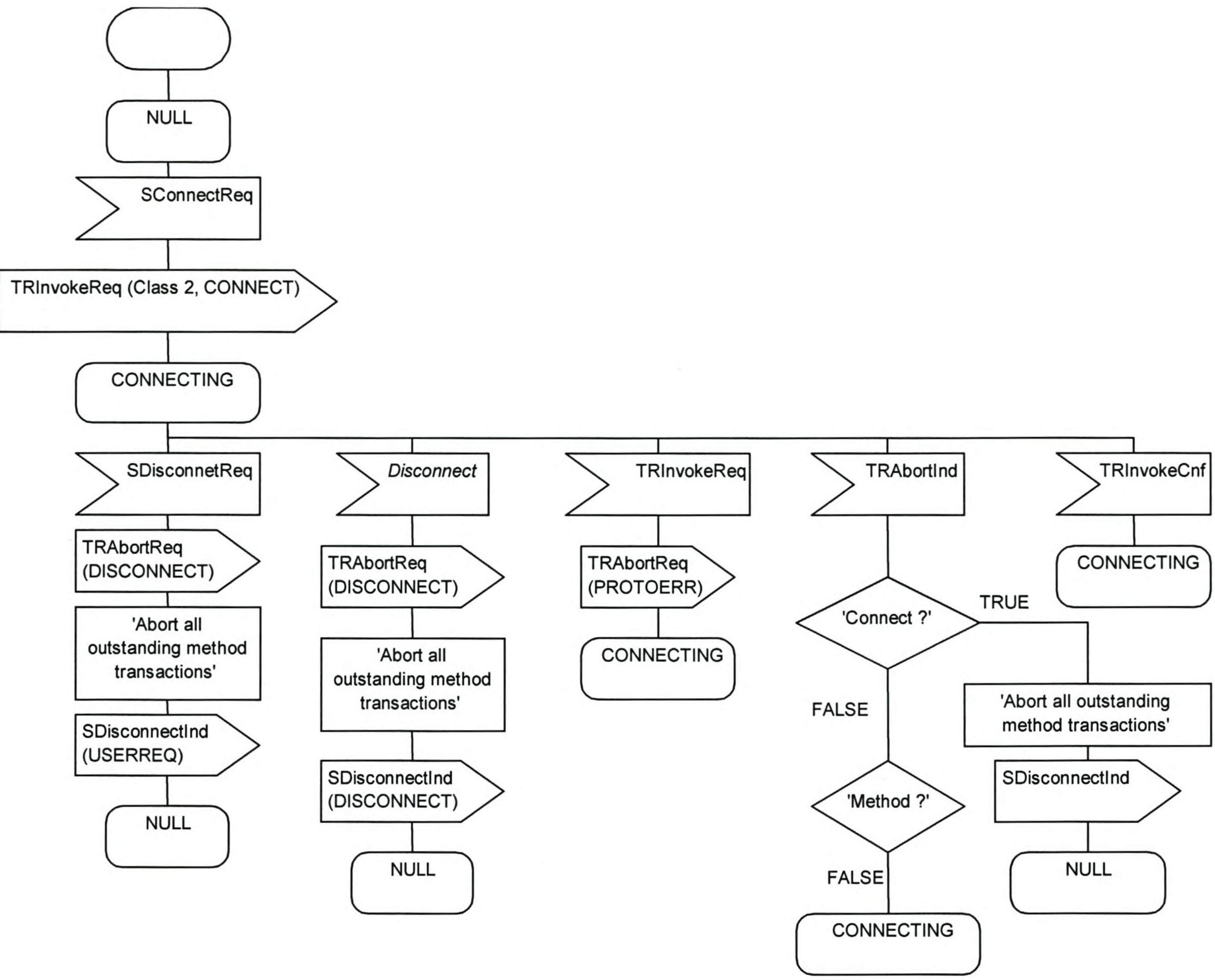

Figure A.44 Client Session Process

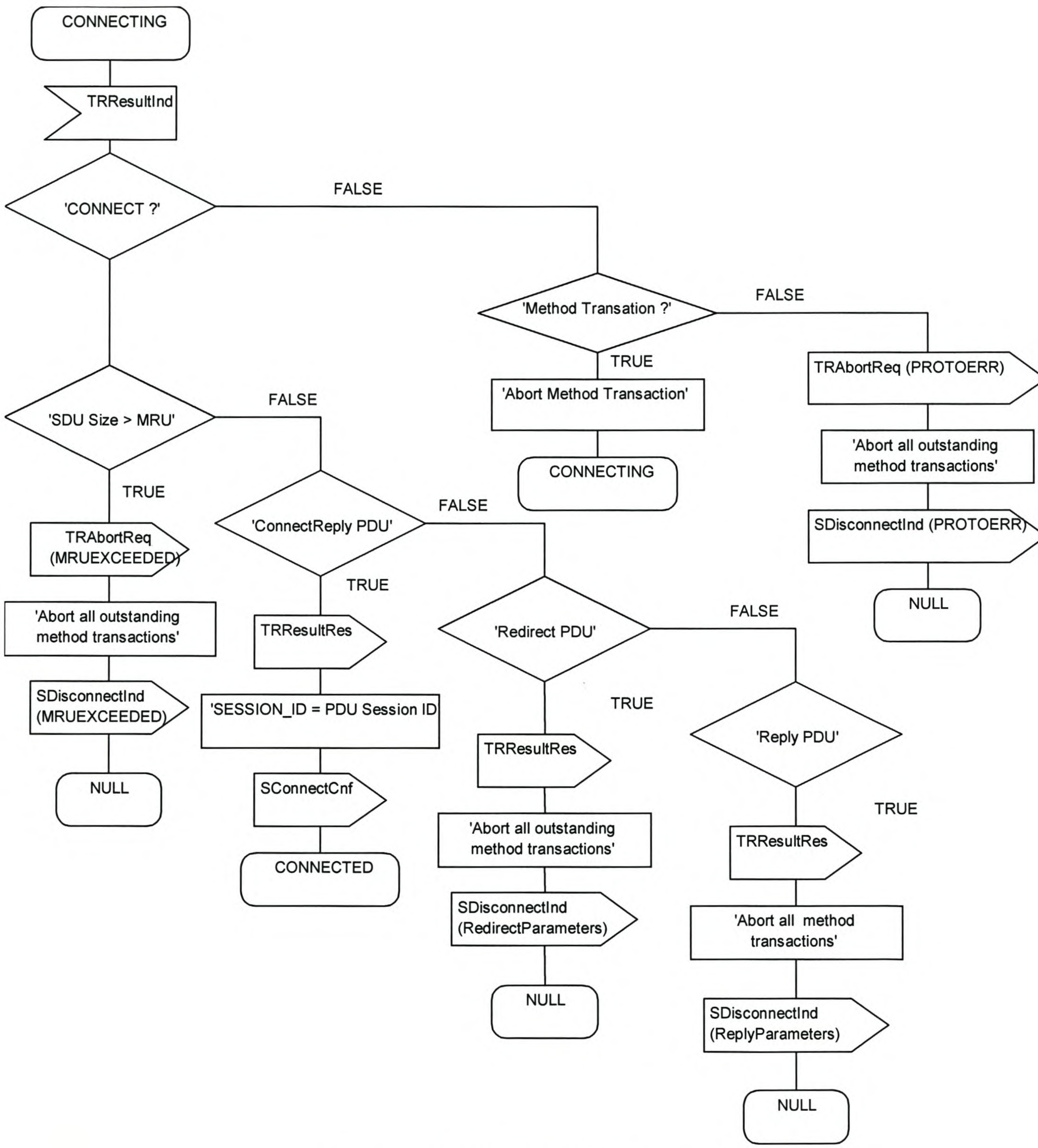

Figure A.45 Client Session Process

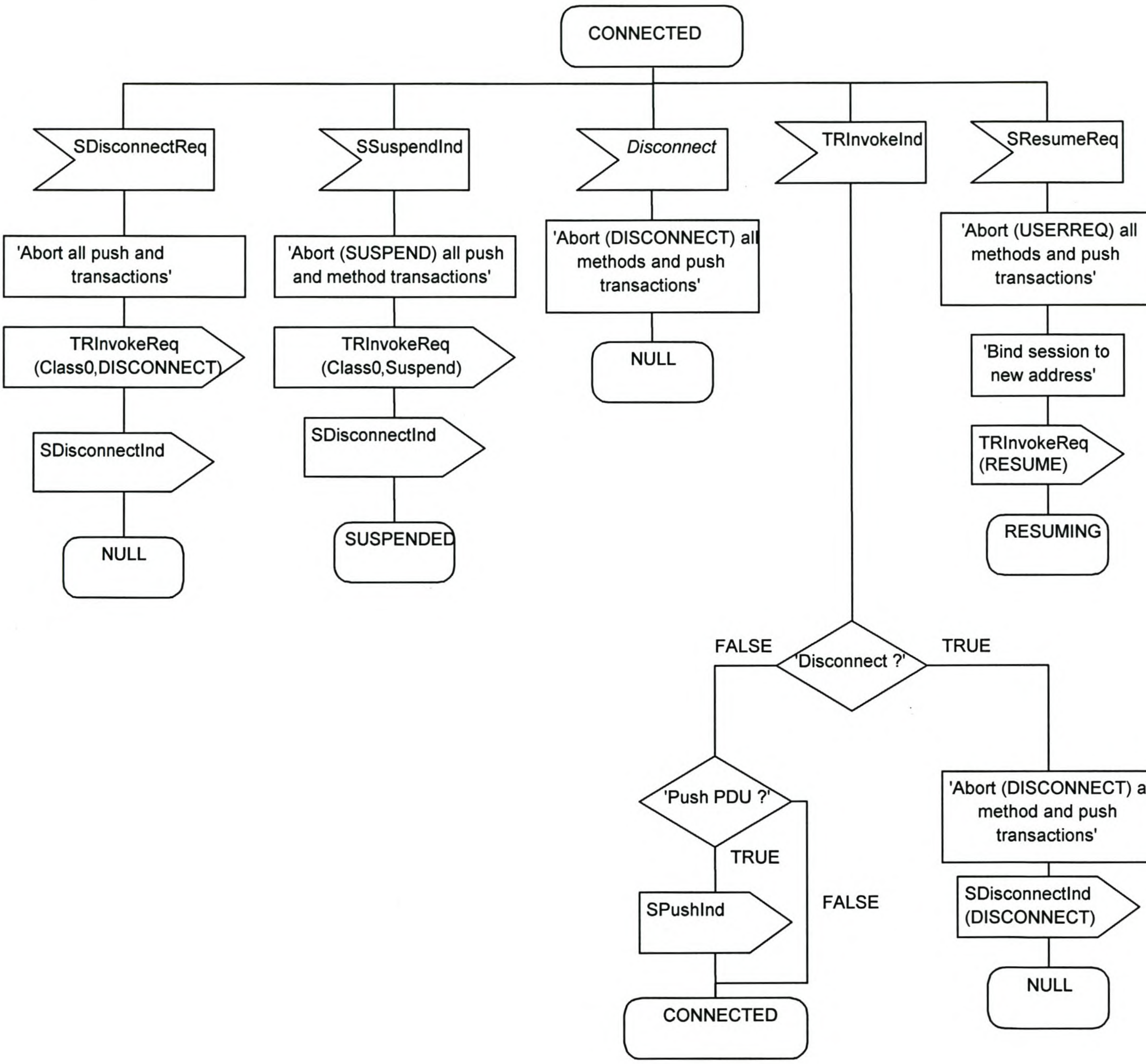

Figure A. 46 Client Session Process

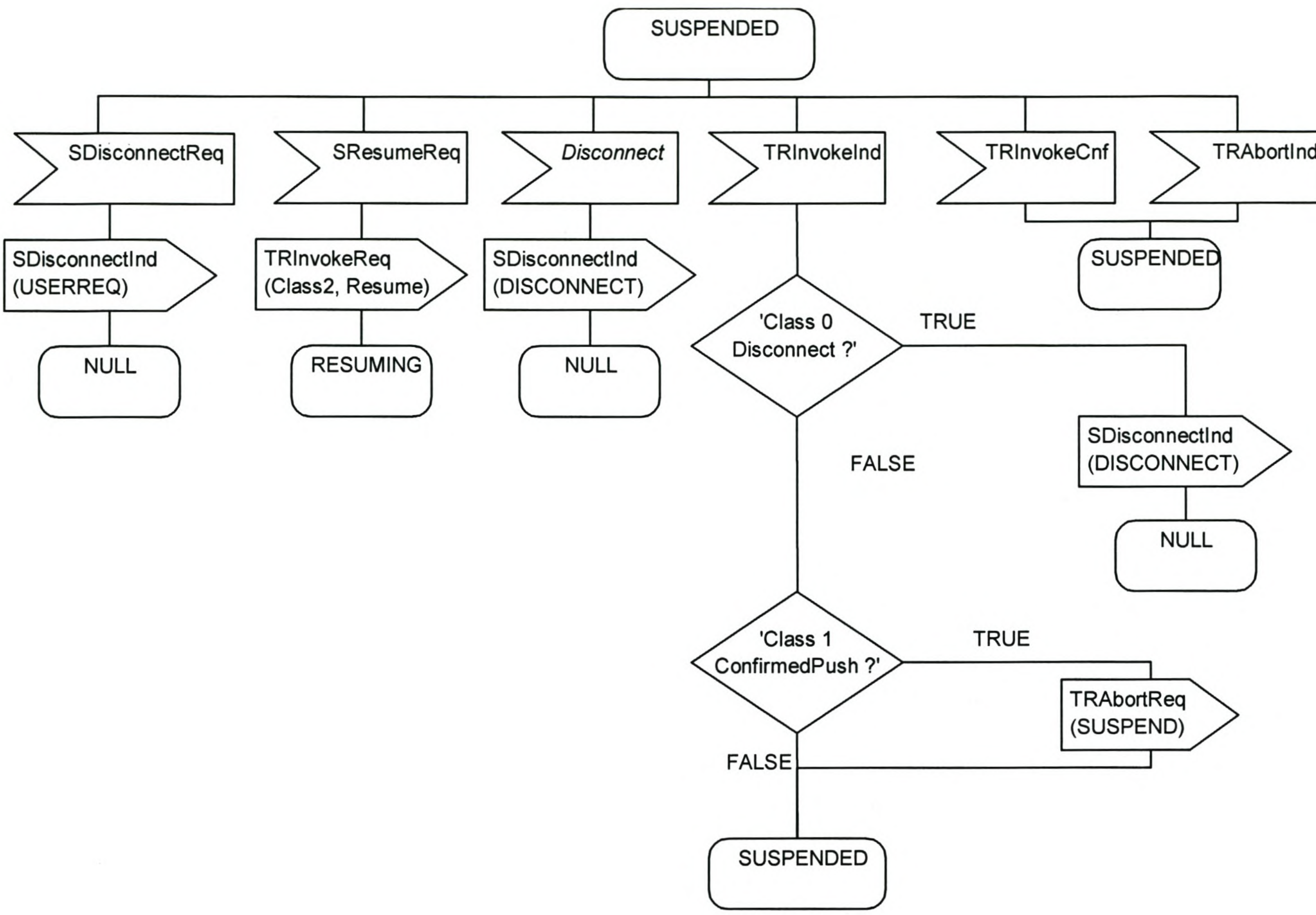

Figure A.47 Client Session Process

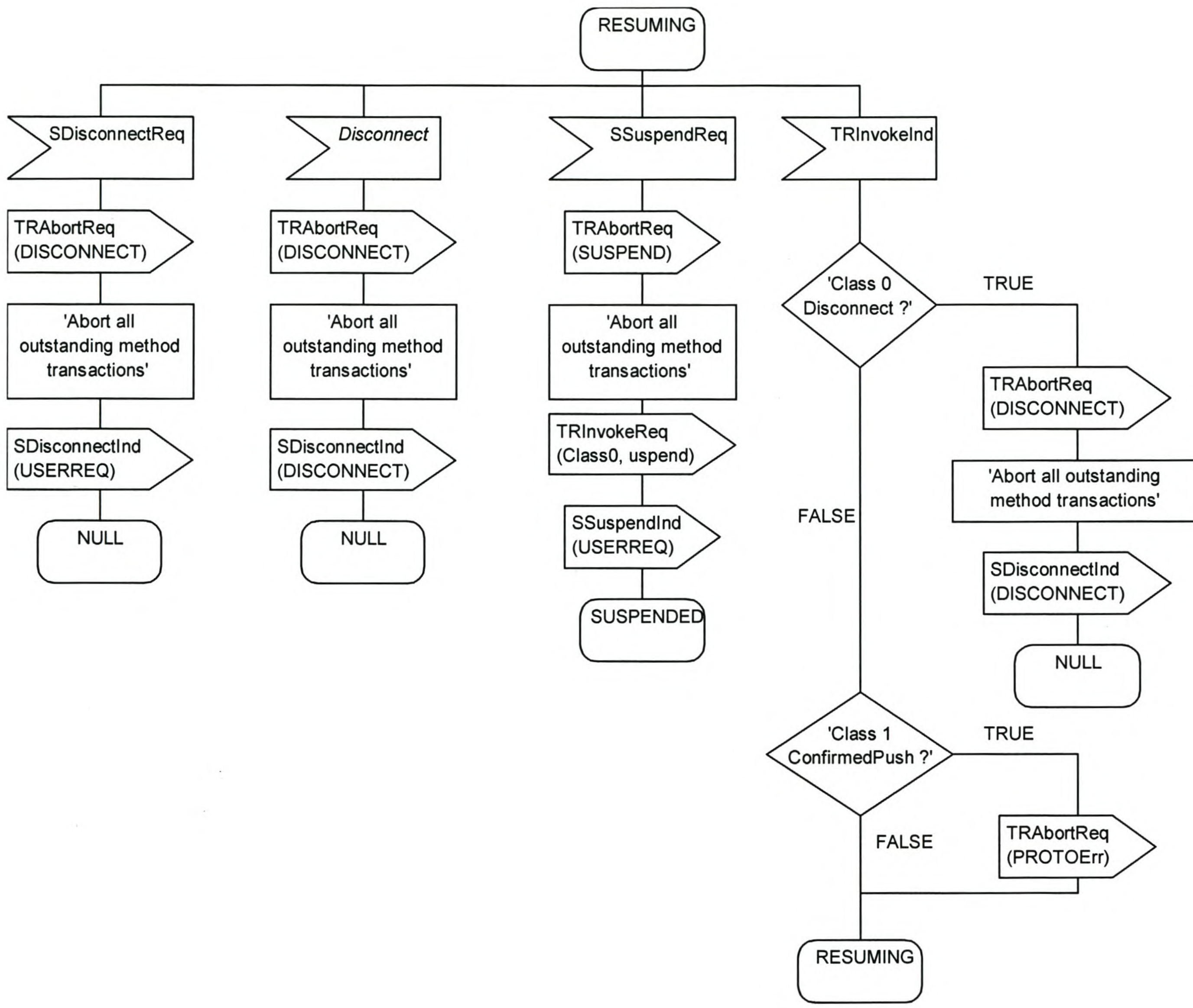

Figure A.48 Client Session Process

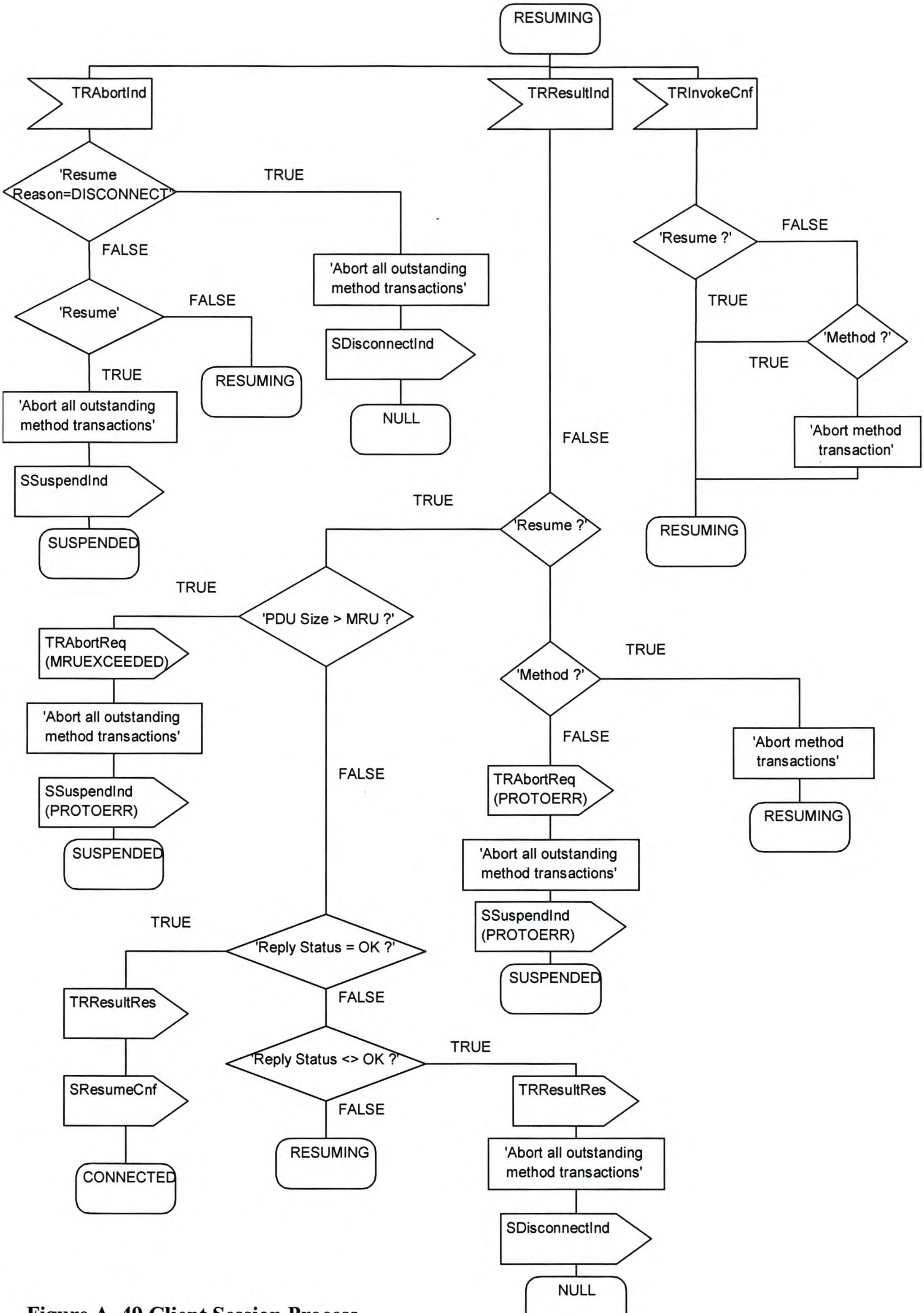

Figure A. 49 Client Session Process

A.2.2.2 Client Push Process (Client)

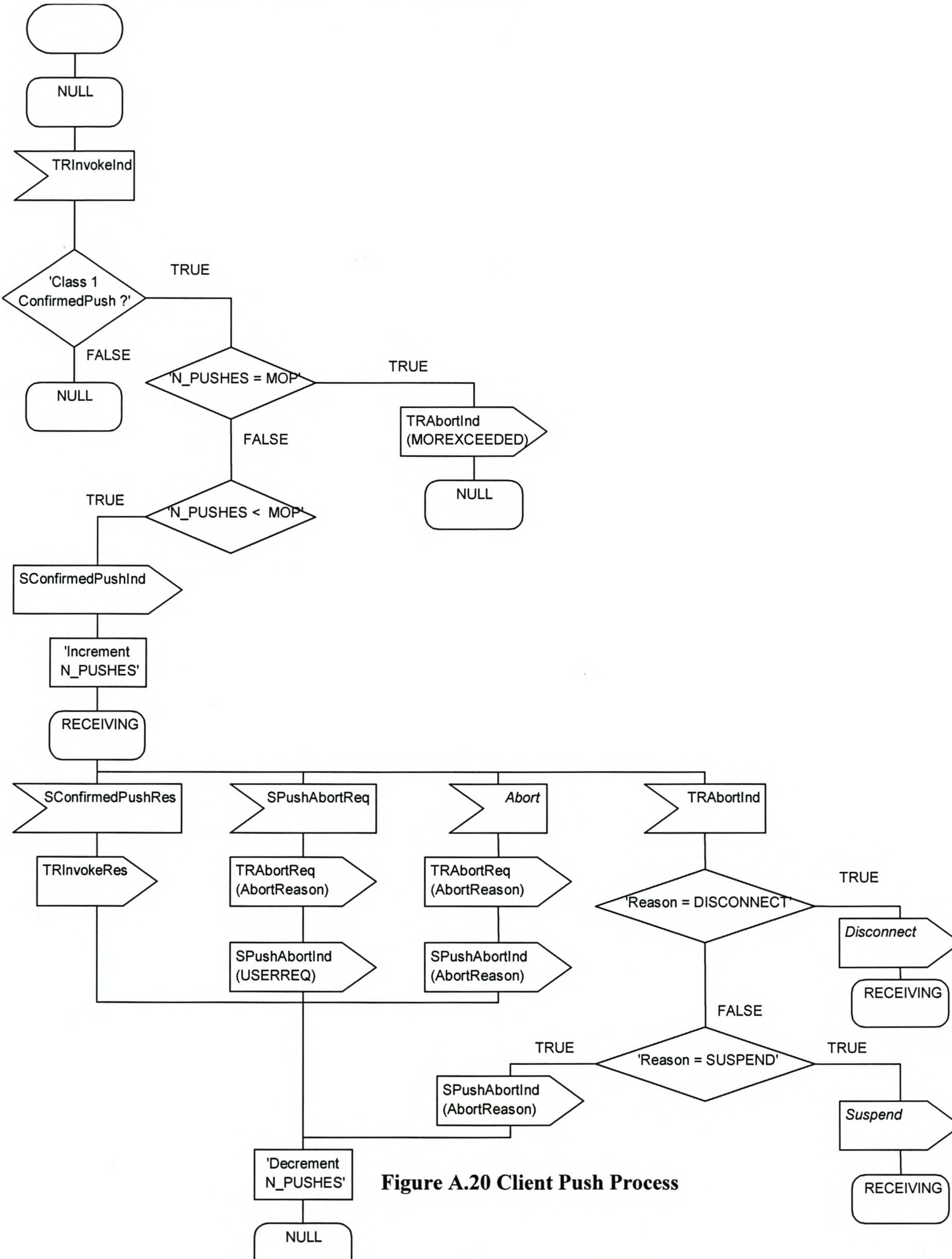

Figure A.20 Client Push Process

A.2.2.3 Client Method Process (Client)

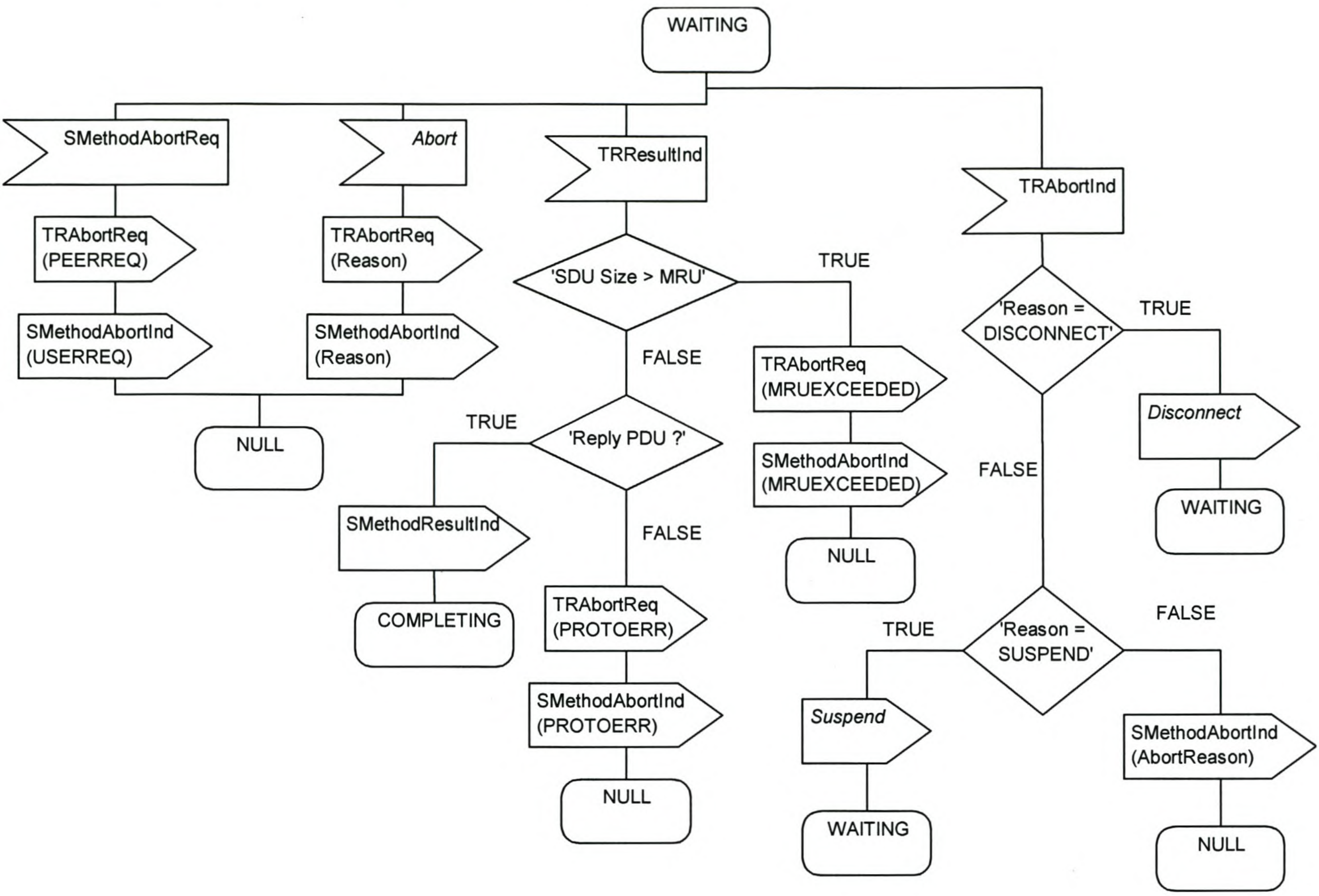

Figure A.50 Client Method Process

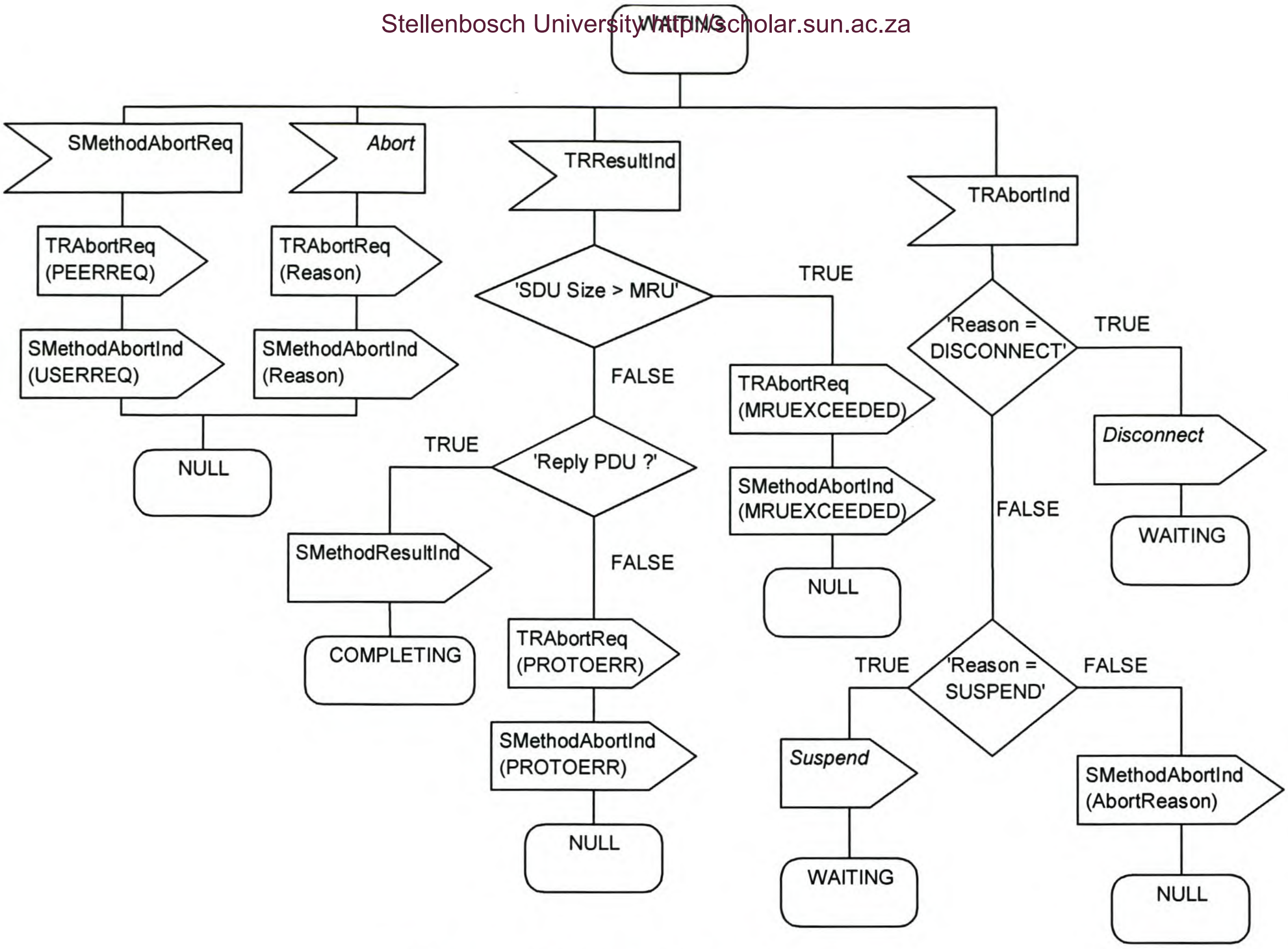

Figure A.51 Client Method Process

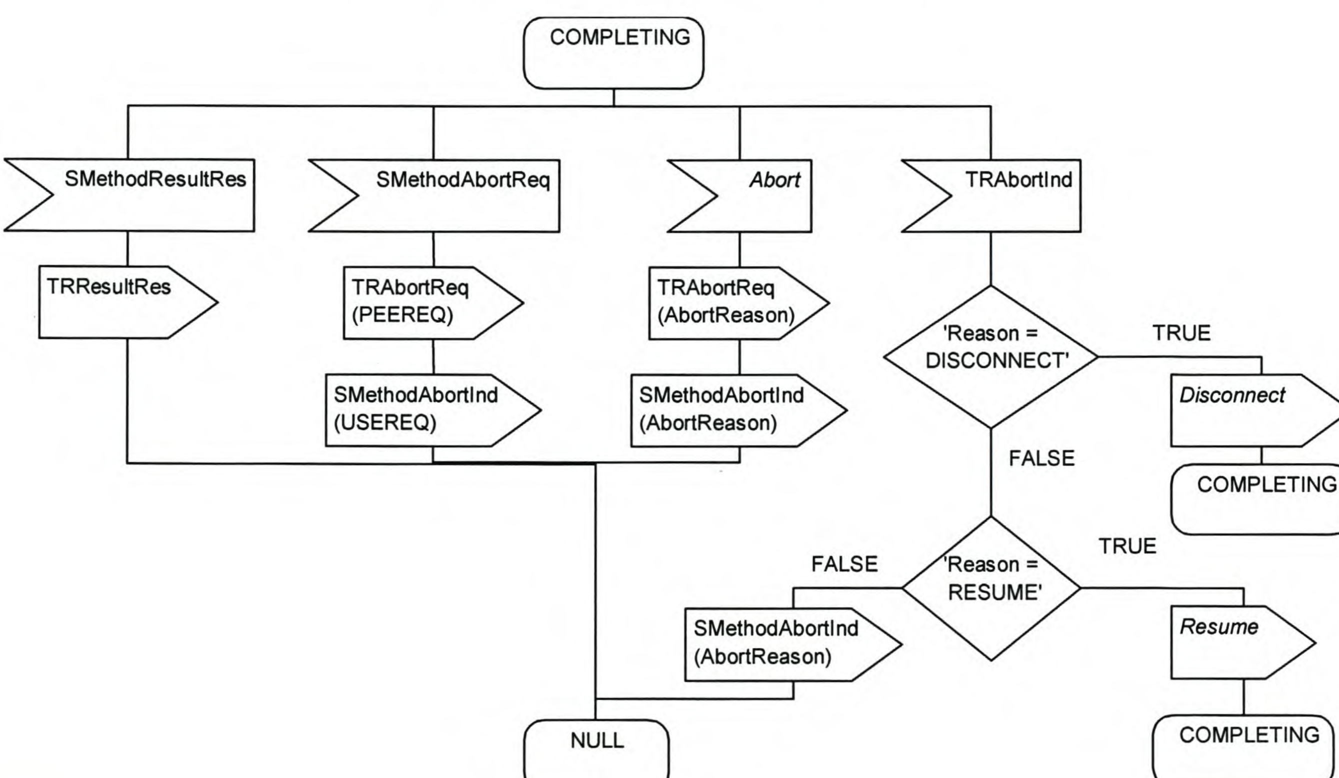

Figure A.52 Client Method Process

Appendix B

GSMFTP Efficiency

The GSMFTP protocol uses a window mechanism. The FTP procedure divides a given file size into a number of fixed length datagrams. A fixed number of datagrams forms a window. Only correctly received windows are acknowledged and not individual datagrams. No acknowledgements are issued for incorrectly received windows. This result in quite a significant saving of bandwidth since one acknowledgement is issued for a number of datagrams, which forms a window.

Figure B1 shows that every GSMFTP window is followed by an acknowledgement. The *sending party* transmits the bulk of the data to the *receiving party*. If the *receiving party* receives a complete and intact window, it sends an acknowledgement to the *sending party*.

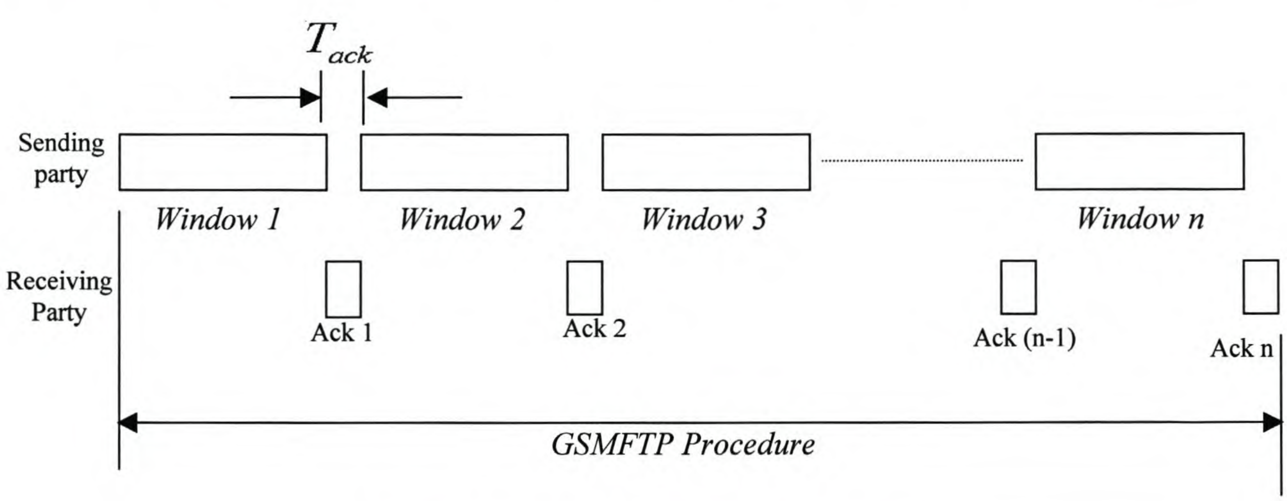

Figure B 1 GSMFTP Window Mechanism

The *sending party* will wait for an acknowledgement from the *receiving party* a maximum period of time. If this maximum period of time expire while the sending party is still waiting for an acknowledgement from the *receiving party*, the sending party goes into a *PAUSE* state. The time that a receiving party must wait for an acknowledgement must be at least the RTT on such a GSM data channel plus any delays due to processing of the information. The transmission delay for data channels on cellular networks is on average 100 ms [42]. In the simulations that were executed, *Tack* was selected as 250 ms (RTT = 200 ms and Processing and other delays = 50 ms).

Figure B2 shows the architecture of one GSMFTP window. One such window consists of a fixed number of N datagrams. A time difference, *Ta*, seperates successive datagrams from each other.

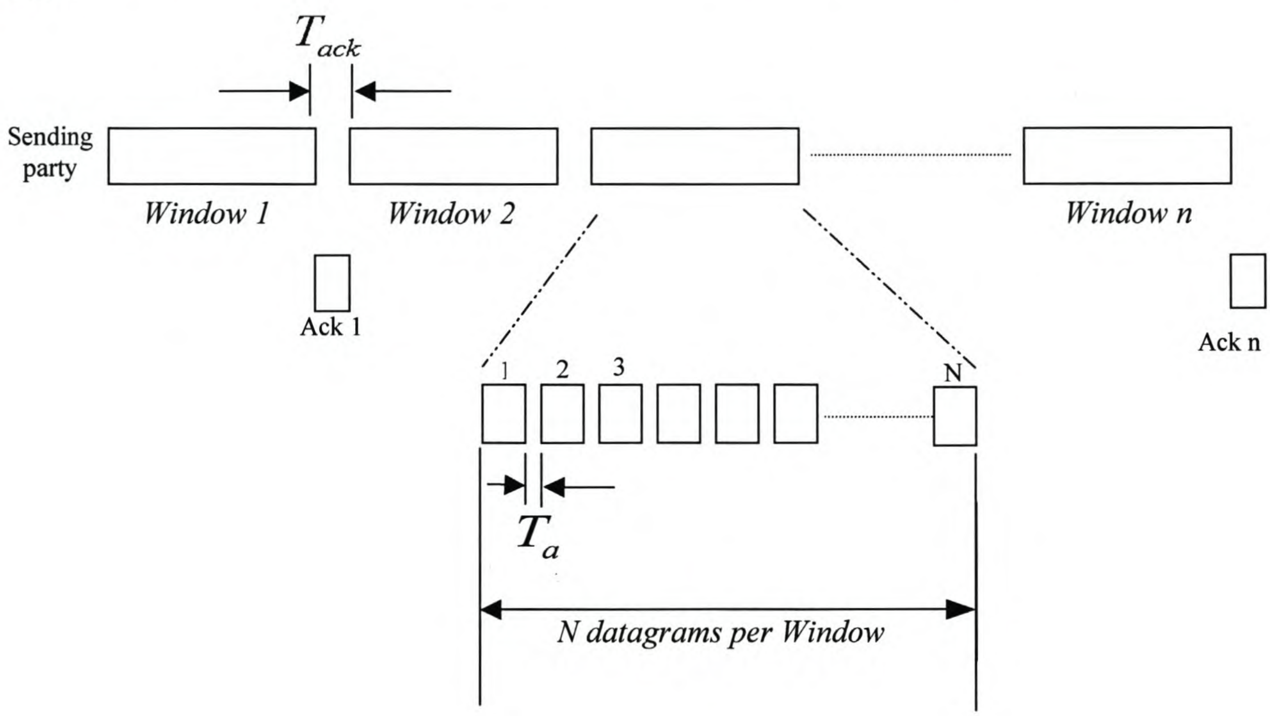

Figure B 2 Architecture of a GSMFTP Window

$\underline{T_a}$ represents the time used for scheduling other tasks and to prepare the next datagram. The time spend on scheduling of other tasks and the preparation of the next datagram is effectively a waste of bandwidth. No data are transmitted on the channel during the T_a and T_{ack} periods.

The efficiency for the GSMFTP procedure may be expressed as

$$\eta = \frac{TotalNoOfWindowsWithValidData}{TotalNoOfWindowsTransmitted}.$$

With

$TotalNoOfWindowsWithValidData$ = Number of windows transmitted if there were no failures
$TotalNoOfWindowsTransmitted$ = Number of windows transmitted including retransmissions.

The following example shows an FTP procedure that must transmit 25 windows of data. For this example

$$TotalNoOfWindowsWithValidData = 25 \text{ and}$$
$$TotalNoOfWindowsTransmitted = 25+6=31$$

which results in an efficiency

$$\eta = \frac{TotalNoOfWindowsWithValidData}{TotalNoOfWindowsTransmitted} = \frac{25}{25+6} = 0.81$$

Figure B3 is an illustration of the example. Six windows are retransmitted. The probability of window failure is

$$P_w = \frac{1}{5} = 0.20$$

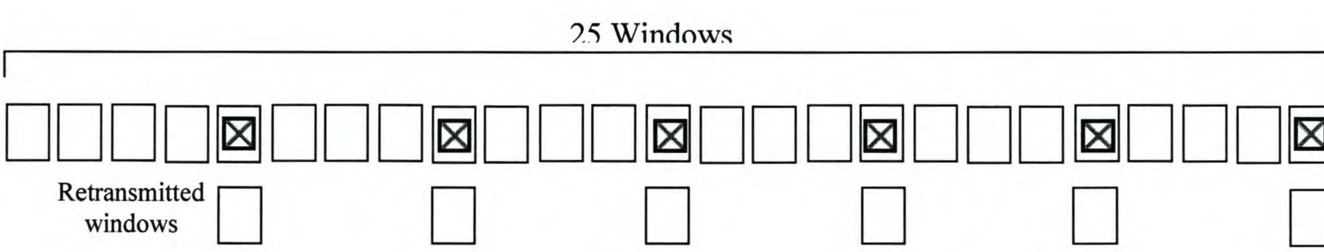

Figure B 3 GSMFTP Example

The probability of window failure Pw = 0.20 means that every fifth window will fail. If the probability of window failure is Pw, then statistically every $\frac{1}{P_w}$ will fail. The efficiency of the FTP procedure when looking at windows as the transport unit is

$$W\eta = \frac{W}{W + \frac{W}{\frac{1}{P_w} - 1}} = \frac{1}{1 + \frac{1}{\frac{1}{P_w} - 1}} = \frac{1}{1 + \frac{P_w}{1 - P_w}} = 1 - P_w$$

The efficiency of one window is defined as

$$D\eta = \frac{ND_n}{N(4 + D_n + T_a B) + T_{ack} B}$$

where

N	is the number of datagrams per window
D_n	is the number of user bytes per datagram
4	is the number of control information bytes per datagram
T_a	is the time between two successive datagrams
B	is the bandwidth of the channel in bytes per second.
T_{ack}	is the time between two successive windows

The combined efficiency of the complete GSMFTP procedure is expressed as

$$GSMFTP_\eta = W_\eta \times D_\eta = \frac{ND_\eta}{N(4 + D_\eta + T_a B) + T_{ack} B}(1 - P_w)$$

The probability of window failure, P_w, may be expressed in terms of the probability of datagram failure P_p

$$P_w = NP_p$$

The final equation expresses the complete GSMFTP efficiency on a channel with bandwidth B bytes per second, fixed windows consisting of N datagrams.

$$GSMFTP_\eta = \frac{ND_\eta}{N(4 + D_\eta + T_a B) + T_{ack} B}(1 - NP_\eta)$$

The optimal window size can be found for fixed datagram size. The solution to finding the optimal efficiency is to differentiate the equation with respect to N.

$$\frac{d}{dN} GSMFTP\eta = \frac{D_n(1-2P_pN)}{N(4+D_n+T_aB)+T_{ack}B} + \frac{ND_n(1-2P_pN)(4+D_n+T_aB)}{(N(4+D_n+T_aB)+T_{ack}B)^2}$$

$$\frac{d}{dN} GSMFTP\eta = \frac{D_nT_{ack}B + 2(4+D_n+T_aB-P_pT_{ack}B) - 3P_p(4+D_n+T_aB)N^2}{(N(4+D_n+T_aB)+T_{ack}B)^2}$$

To find the optimal efficiency the null points to the above equation must be found.

Appendix C
ORBCOMM Detail

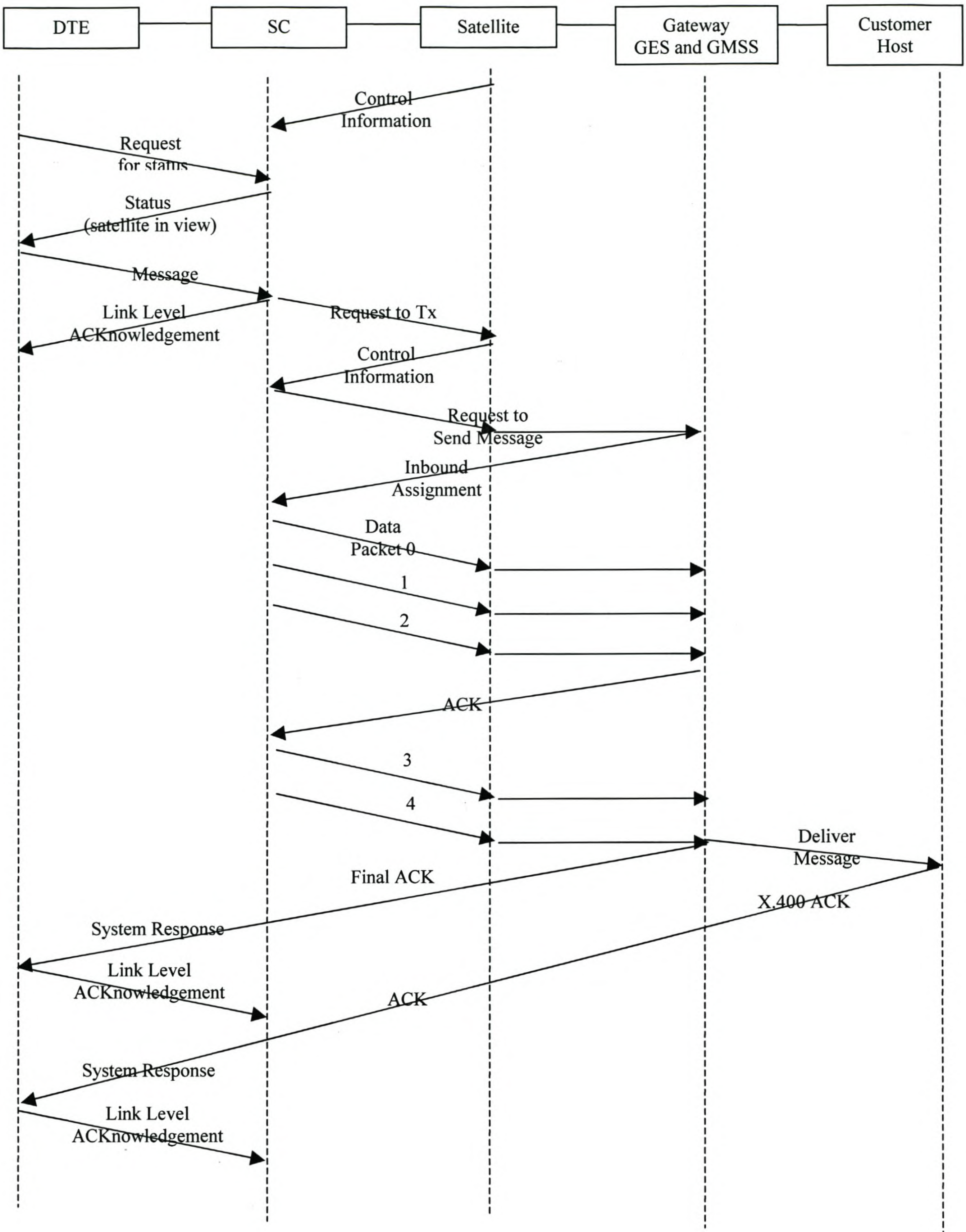

Figure D.1 SC Originated Data Transfer

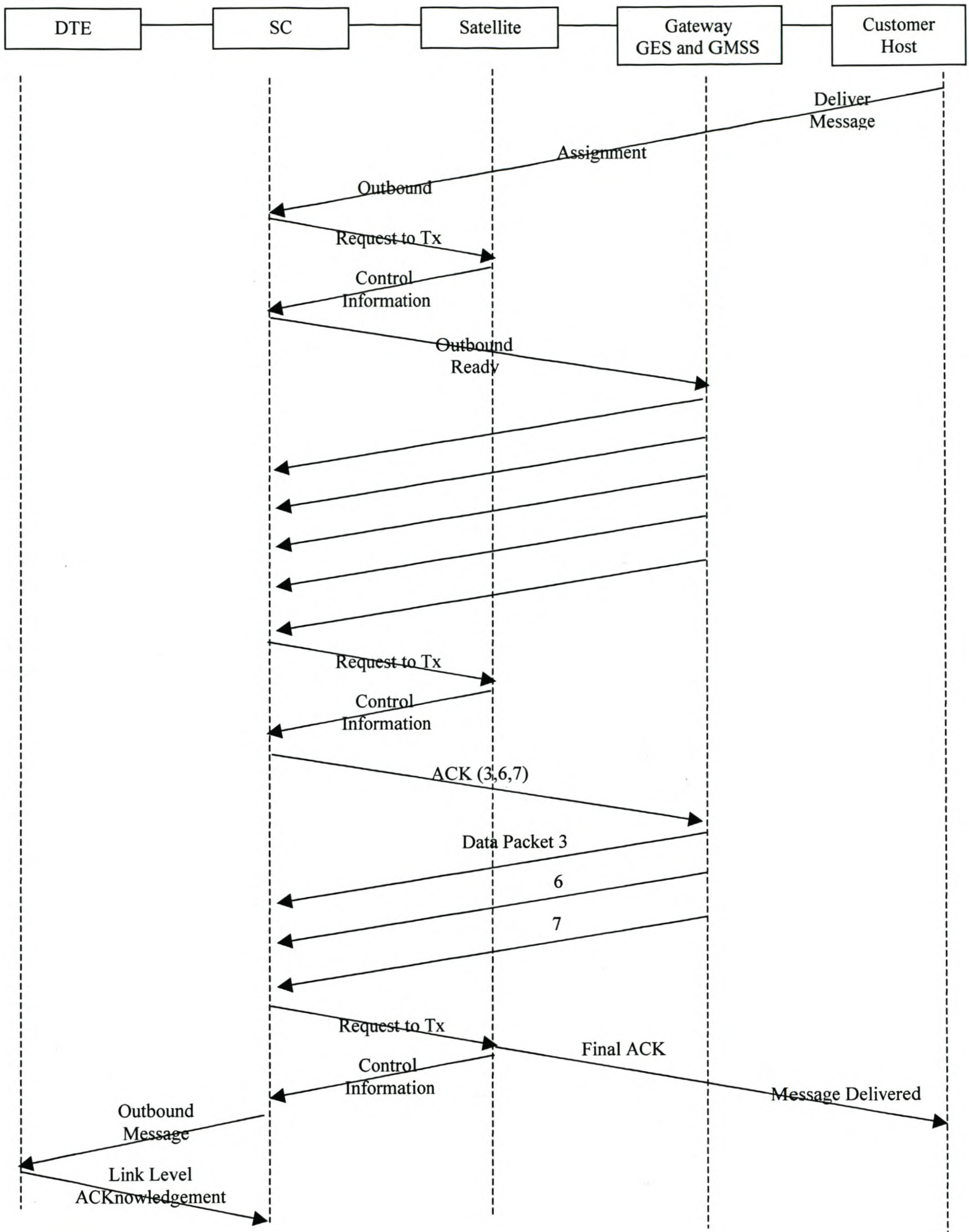

Figure D.2 SC Terminated Data Transfer