

ecai

**Universidade Federal de Santa Catarina
Centro Tecnológico
Curso de Engenharia de Controle e
Automação Industrial**

ufsc

***Desenvolvimento de Padrões para Teste de
Software (CNCS e Aegis) de Gerência
de Redes VSAT***

*Monografia submetida à Universidade Federal de Santa Catarina
como requisito para a aprovação da disciplina:*

EEL 5901: Projeto de Fim de Curso

Rodrigo Milani Medeiros

Florianópolis, Fevereiro de 1998

**Desenvolvimento de Padrões para Teste de Software
(CNCS e Aegis) de Gerência de Redes VSAT**

Rodrigo Milani Medeiros

Esta monografia foi julgada no contexto da disciplina
EEL 5901: Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação Industrial

Banca Examinadora:

Marie Creque
Orientador Empresa

Marcelo Ricardo Stemmer
Orientador UFSC

Prof. Augusto Humberto Bruciapaglia
Responsável pela disciplina e Coordenador do Curso

Prof. Joni da Silva Fraga, Avaliador

Gean Carlo Dallagnolo, Debatedor

Hélio S. Araújo, Debatedor

Resumo

Este trabalho descreve a revisão da metodologia de testes aplicado nas novas versões de software de gerenciamento de redes VSAT. Estas redes consistem de uma estação central que conecta via satélite estações remotas equipadas com antenas parabólicas de pequeno porte - diâmetro inferior a 2.4 m.

A metodologia de teste corrente tornou-se ineficiente devido ao crescimento do número de protocolos de acesso à rede. A complexidade dos softwares para gerenciamento deste tipo de rede aumenta proporcionalmente a este número de protocolos, exigindo procedimentos de teste cada vez mais extensos.

Realizou-se um estudo detalhado do processo em questão para definir um teste de regressão que minimizasse o número de defeitos encontrados após o lançamento do software no mercado. Dada a impossibilidade de se testar todas as funcionalidades do sistema em questão, definiu-se um procedimento para simular situações mais comuns no uso prático, dentro de um limite de tempo pré estabelecido. O sistema, ou objeto de teste, leva o nome de CNCS.

Como recursos de estudo do processo foram utilizados a base de dados de defeitos reportados do CNCS, entrevistas com os responsáveis pelo desenvolvimento do sistema e uma experiência prática em uma central de controle de redes VSAT.

Abstract

This report describes the review of the methodology used for testing the new versions of VSAT network management software. This networks have a central station that connects remote stations through satellite. The remote stations are equipped with small-size satellite dishes - under 2.4 m.

The current methodology has become inefficient due to the growth of the number of access protocols supported within this networks. The software complexity increases proportionally with this growth, demanding more extensive test procedures.

An comprehensive study of the process has been performed so as to define a regression test that minimizes the number of defects (bugs) found after launching the software into the market. As it is not possible to test all the software functionality, a procedure was defined to simulate the most common situations in the practical use of CNCS - the tested system.

The following resources were used to study the process: the database of defects reported in CNCS, interviews with software developers and a practical experience in a VSAT network operations center.

Sumário

1. Introdução.....	6
2. Redes VSAT.....	8
2.1. A rede Clearlink, fornecida pela AT&T Tridom.....	8
2.2. Topologia da rede.....	8
2.3. Comutação de pacotes.....	10
2.4. Configuração e Controle da Rede.....	10
2.5. Componentes da Rede Clearlink (Clearlink Network Resources).....	11
2.6. Sistema de Controle da Rede Clearlink.....	12
2.7. O Software CNCS - Clearlink Network Control System.....	14
2.7.1. Menu de Configuração.....	14
2.7.2. Menu de Status / Monitoração.....	14
2.7.3. Menu de Estatísticas.....	14
2.7.4. Menu de Manutenção.....	15
2.7.5. Operação do CNCS.....	15
2.8. Conclusão.....	16
3. Formalização do Problema e Técnicas de Solução: Teste de Regressão do CNCS.....	17
3.1. Introdução.....	17
3.2. Verificação e Validação de Software.....	17
3.3. Metodologia de teste anterior ao projeto.....	18
3.4. Definição do Escopo do teste.....	19
3.5. Metodologia para encontrar a solução.....	20
3.6. Conclusões.....	21
4. Elaboração e Implementação do Plano de Testes.....	22
4.1. Introdução.....	22
4.2. Cronograma utilizado.....	22
4.3. Coleta de dados para desenhar a solução.....	23
4.3.1. Análises dos relatórios de defeito (bug reports ou FOPs).....	23
4.3.2. Entrevistas com desenvolvedores de software.....	26
4.3.3. Verificação de configuração através de comandos ODBC.....	27
4.3.4. Estudo da automação dos testes.....	29
4.3.5. Estágio em um Centro de Operações da Rede - Hub Atlanta, GA.....	29
4.4. Conclusões relativas à elaboração do teste e implementação.....	30
4.4.1. Sobre as entrevistas com os programadores.....	31
4.4.2. Sobre o estágio no NOC de Atlanta, GA.....	31
5. Resultados.....	33
5.1. Descrição do Plano de Testes Proposto.....	33
5.2. Documentação do teste proposto.....	33
5.3. Teste preliminar e teste do CNCS 10.0.....	34
6. Conclusões e Perspectivas.....	36
7. Bibliografia.....	37
8. Anexos.....	39
8.1. Anexo 1: Configurações utilizadas no plano de testes final.....	39
8.2. Camadas OSI na rede Clearlink.....	42
8.3. Anexo 3. Visão geral dos processos e suas localização na arquitetura Clearlink.....	43

Índice de Figuras

figura 1-1: Equipamento VSAT	6
figura 2-1: Arquitetura de uma rede VSAT	8
figura 2-2: Rede VSAT Clearlink, topologia “estrela”	9
figura 2-3: Topologia estrela com <i>broadcast</i> do <i>hub</i> para os VSATs.....	9
figura 2-4: Ligação dos POPs com o Hub central.....	10
figura 2-5: Arquitetura de um HUB	11
figura 2-6: Hierarquia da rede Clearlink	12
figura 2-7: Sistema de menus do CNCS	13
figura 2-8: Base de dados CNCS e nó remoto.....	16
figura 4-1: Tabelas com os todos os recursos da rede Clearlink.....	27
figura 4-2: Resultados de um comando ODB	28
figura 4-3: Tela de configuração de uma porta LAPB	28
figura 4-4: Cenário típico de uma divisão de um NOC	30

1. Introdução

O termo "VSAT" (*Very Small Aperture Terminal*, ou Terminal de abertura muito pequena) se refere ao uso de antenas de pequeno porte para comunicação via satélite de dados, vídeo ou voz. Estas antenas parabólicas têm diâmetro de 0.95 m a 2.4 m, ocupando portanto pouco espaço e tendo como característica a facilidade de instalação.

Na figura 1-1 pode-se ter uma noção de tamanho do equipamento VSAT comparado com uma caneta comum.

As principais vantagens sobre o uso de linhas telefônicas é a rapidez de comutação, rapidez de instalação e indiferença quanto a posição geográfica dentro da área de cobertura do satélite utilizado. Esta última característica é bastante favorável para redes *on-line* de transações bancárias ou de cartões de crédito. Os Bancos do Brasil e Bradesco, por exemplo, utilizam a tecnologia VSAT em boa parte de suas redes.

Em casos onde se torna inviável o uso de linhas telefônicas em locais remotos, a solução VSAT apresenta bom custo-benefício. Um exemplo disto é o monitoramento de oleodutos ou gasodutos de grande extensão. Ao longo destes dutos, os controladores e coletores de dados locais são conectados em tempo real, via satélite, com um centro de controle e operações que controla os nós da rede VSAT.

As redes VSAT vêm expandindo exponencialmente no Brasil. A qualidade, custo e acesso nem sempre adequados à conexões por linha telefônica comum traz a vantagem competitiva a esta arquitetura de rede.

A rede VSAT Tridom tem arquitetura de base de dados e gerenciamento centralizados. Todas as conexões entre os VSATs e seu respectivo computador central são feitas em um concentrador (HUB). Estes Hubs são instalações de tamanho proporcional ao número de VSATs na rede. O Hub Tridom em Atlanta - GA, serve em torno de 10 mil *sites* ou VSATs, tendo em torno de 20 operadores trabalhando 24h/dia.

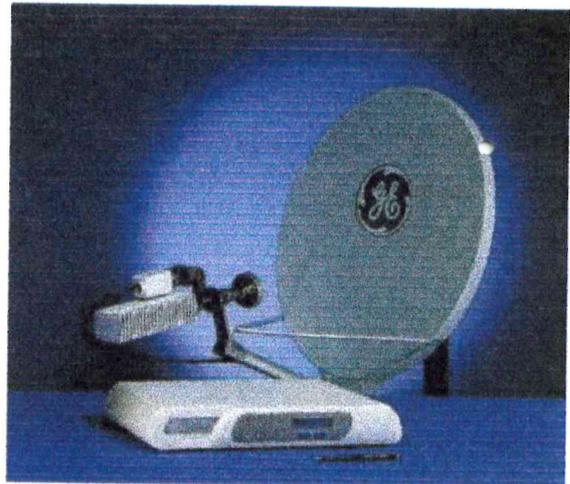


figura 1-1: Equipamento VSAT

CNCS e Aegis são *softwares* de gerência de redes VSAT. A plataforma para o CNCS é um computador tolerante a falhas (SFT - Stratus) com sistema operacional proprietário VOX. O Aegis complementa o CNCS com relação à interface gráfica, em ambiente X-Windows / SunOS4. [Tridom 01]

O objetivo do projeto é estabelecer uma rotina formal de teste para novas versões dos softwares citados, para que se detecte em laboratório os problemas que atualmente são encontrados pelo usuário-cliente. Como grande parte dos clientes são internacionais, a redução de custos é sensível quando os problemas são resolvidos em laboratório, antes do lançamento das novas versões no mercado.

As rotinas de testes deverão envolver a configuração de elementos da rede Tridom, em ambiente de laboratório com cenários semelhantes ao encontrado em um centro de operações (Hub) real.

Quanto mais abrangente for o teste, menor a probabilidade de falhas posteriores. Contudo, torna-se inviável testar-se **todas** as configurações possíveis devido à complexidade do software e quantidade de protocolos envolvidos. Um dos desafios, portanto, é definir um escopo ótimo para os testes. Atualmente, as novas versões são instaladas no laboratório da engenharia e utilizado durante as atividades normais de laboratório. Este método não tem se mostrado eficiente, pois poucas são as configurações utilizadas.

No desenvolvimento deste projeto foi necessário o entendimento extensivo da arquitetura de rede VSAT, dos *softwares* de gerenciamento em questão e das configurações possíveis nas diferentes aplicações. As atividades desenvolvidas incluíram cursos de treinamento, experiência prática em um centro de operações de rede, entrevistas com analistas de sistema, análises de relatórios de defeitos e testes em laboratório.

A rede VSAT - Tridom suporta a conexão entre redes de diferentes protocolos, com diferentes aplicações e em diversas plataformas. O presente projeto trouxe conhecimento na área de hardware e software em redes de computadores de comunicação de dados, gerência de sistemas multiprocessados, protocolos de comunicação mais utilizados na indústria, além de experiência prática na padronização de rotinas em um ambiente de engenharia.

O presente trabalho foi desenvolvido na empresa GE Spacenet, do grupo General Electric que assumiu em 1º de julho de 97 o controle da então AT&T Tridom.

Com objetivo de proporcionar uma visão geral da arquitetura da rede em questão, temos no capítulo 2 uma breve descrição de seus componentes básicos. O capítulo 2 traz também alguns conceitos de gerenciamento de redes VSAT e discute aspectos dos softwares que são objeto de estudo neste trabalho: CNCS e Aegis.

O capítulo 3 pretende discutir questões referentes à metodologias de teste em geral, ressaltando as técnicas utilizadas na rotina e teste preliminar e final.

O capítulo 4 descreve a implementação da rotina de teste proposta, discutindo os problemas encontrados e as soluções adotadas.

O capítulo 5 traz os resultados obtidos após os testes preliminares e após a execução do teste final.

O capítulo 6 expõe as conclusões obtidas, apontando as mudanças desejáveis para melhorar o processo no futuro.

2. Redes VSAT

2.1. A rede Clearlink, fornecida pela AT&T Tridom

“*Clearlink Network*” é um termo genérico que se refere à classe de redes de informação baseados no equipamento fornecido pela AT&T Tridom. Estas redes são baseadas na conexão por satélite entre uma estação central (Hub Station) e um grande número de antenas pequenas, de baixa potência, que são os VSATs. Observe a figura 2-1.

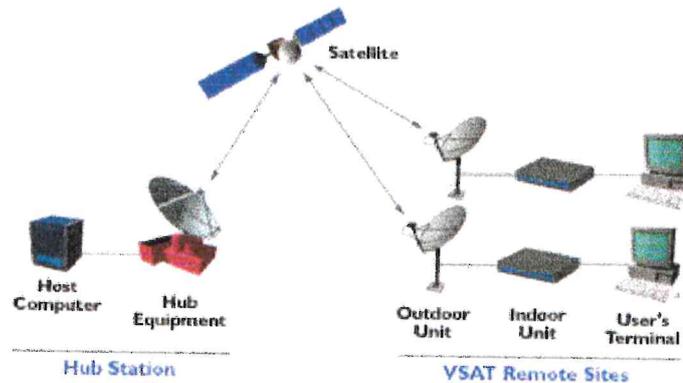


figura 2-1: Arquitetura de uma rede VSAT

A antena na estação central tem diâmetro superior a 10m, enquanto nas estações VSAT, o diâmetro varia de 0.9 a 2.4m. O VSAT transmite normalmente menos de 2W de potência.

A rede Clearlink foi projetada inicialmente como uma rede de dados digitais. A aplicação típica é para suprir as necessidades de comunicação entre filiais de grandes corporações ou redes de venda a varejo, onde as várias filiais distribuídas geograficamente possuem equipamento de processamento de dados que precisa comunicar-se com a central periodicamente com a sede matriz.

A instalação nos escritórios remotos é facilitada devido ao tamanho compacto da antena VSAT. Já a estação central (Hub Station) é em geral instalada nas proximidades da sede matriz ou centro de processamento de dados.

2.2. Topologia da rede

A rede VSAT AT&T Tridom (Clearlink) suporta comunicação bidirecional entre a estação central e os VSATs. Cada VSAT pode transmitir para a estação central e a estação central pode transmitir para cada VSAT específico. Dois VSATs não podem se comunicar diretamente. A topologia de rede é, então, tipo “estrela”, conforme mostrado na figura 2-2:

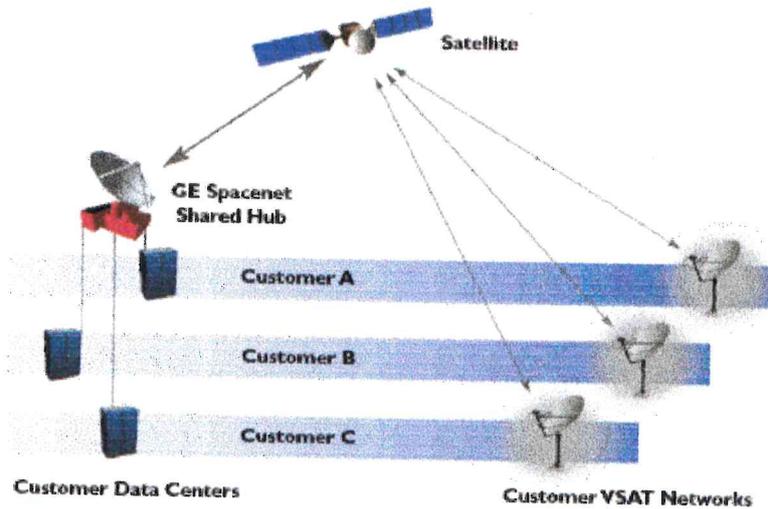


figura 2-2: Rede VSAT Clearlink, topologia “estrela”

A comunicação do *hub* para os VSATs via satélite é do tipo “broadcast”, ou seja, o Hub pode enviar dados a todos os VSATs simultaneamente. Os dados vindos do *hub* são recebidos por todos os VSATs simultaneamente. Na comunicação entre o *hub* e um VSAT específico, o pacote de dados (*data packet*) contém o endereço único da estação VSAT em questão. A figura 2-3 ilustra a topologia estrela com a característica de *broadcast*.

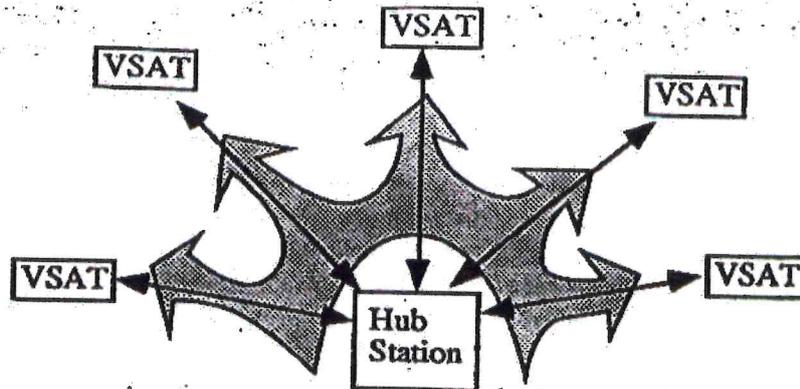


figura 2-3: Topologia estrela com *broadcast* do *hub* para os VSATs

A rede em questão não é puramente baseada em satélite, já que suporta também a comunicação de dados via terrestre, através de linhas telefônica ou outros meios. Conexões (*links*) terrestres são frequentemente usados quando o *hub* não está no mesmo local que o centro de dados ou sede matriz. Para estes casos uma Interface entre o computador com a base de dados central e a rede é fornecida (*host interface*). A *Host Interface* (HI) fica no centro de dados e é conectada ao *hub* através de uma conexão terrestre.

A topologia estrela é válida ainda com a conexão terrestre, apesar de o meio de comunicação ser diferente. O *hub* pode suportar várias *Host Interfaces*. Contudo, o *hub* não tem capacidade de *broadcast* para as HIs.

2.3. Comutação de pacotes

Os VSATs e as HIs são conectados por comutação de pacotes através do *hub*. A comutação de pacotes é feita por um “sistema chaveador” (Switching System, SS ou simplesmente switch), que fica no *hub* e é o nó central da rede. O *Switching System (SS)* determina a rota para pacotes que circulam entre os nós da rede (VSATs e HIs).

Para dar suporte a diversos clientes em um mesmo HUB, pode-se conectar os computadores centrais de cada cliente a um dos pontos de presença do HUB (POPs). Assim, a base de dados da VISA cartões de crédito pode estar em Los Angeles, conectado ao POP local por via terrestre que faz a ligação com o HUB. Uma vez no HUB, todos os VSATs espalhados pelos EUA podem acessar a base da VISA para transações de cartão de crédito. A figura 2-4 ilustra a conexão dos POPs com o HUB de Atlanta / Marietta nos EUA .



figura 2-4: Ligação dos POPs com o Hub central.

2.4. Configuração e Controle da Rede

Até agora apresentou-se três tipos de nós da rede VSAT: A *Host Interface (HI)*, O *Switching System (SS)* e a estação VSAT. Existe também um quarto tipo de nó de rede que é o Computador de Controle da Rede (NCC - *Network Control Computer*).

O NCC não é propriamente parte da rede, já que não existe tráfego de dados do cliente final por este nó. A Função do NCC é de gerenciamento da rede (Network Management), que inclui o processamento de alarmes e alertas dos demais nós de rede, coleta de estatísticas da rede, fornecer os arquivos de sistema para nós remotos e geração de comandos de configuração e manutenção necessários para o gerenciamento da rede.

O aplicativo executado no Network Control Computer (NCC) é o CNCS, objeto de teste no presente relatório.

O NCC é conectado diretamente ao(s) *Switching System(s)* em geral por conexões terrestres (linhas telefônicas dedicadas). O NCC é um computador tolerante a falhas (SFT - Stratus) com sistema operacional proprietário VOX. Terminais e estações Unix podem ser conectadas ao NCC, para servir como interface para os pessoal de operações e facilitar o monitoramento e controle da rede.

Uma rede *Clearlink* potencialmente pode ter vários switches interconectados por linhas terrestres de alta velocidade, controlados por um ou vários NCCs.

Na prática temos em geral um *switch* para uma ou várias partições de rede (clientes) controlado por um NCC dedicado para cada partição ou dividido entre várias partições.

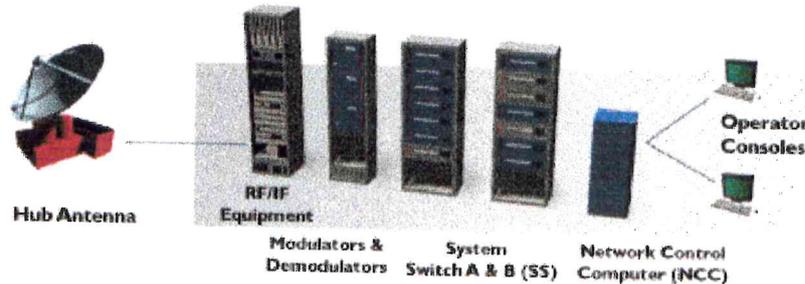


figura 2-5: Arquitetura de um HUB

No Hub de Atlanta + Marietta, Georgia, opera-se com dezenas de redes (partições - clusters), 15 *switches* e 2 NCCs divididos entre as partições. O número de VSATs controlado por este centro de operações é 16000, distribuídos nos países da América Central, do Norte e do Sul.

2.5. Componentes da Rede Clearlink (Clearlink Network Resources).

Todos os componentes de hardware e software da rede Clearlink são organizados em uma estrutura hierárquica e em camadas. Abaixo temos um diagrama desta hierarquia:

- | | |
|------------------------------------|--|
| - Node | - Nó (VSAT, HI, SS ou NCC) |
| -- NPA (Nodal Point of Attachment) | - Ponto de conexão no nó (associado a um processador) |
| --- Task | - Tarefa (ex: Sistema operacional, driver para X.25) |
| ---- Port | - Porta (Ex: Portas seriais ou ethernet) |
| ----- Link | - Enlace (Para protocolos com enlace) |
| ----- Link Station | - Estação de Enlace (ex. Voice Module ou estação VDLC) |

Cada nó na rede possui um endereço único. Um nó é constituído de NPAs, tarefas, portas, enlaces e estações de enlace. *Switching Systems* (SS) e *Host Interfaces* (HI) são multiprocessados, e cada placa com um processador corresponde a um NPA dentro de um nó. O VSAT possui um só processador, e por convenção o número deste NPA é sempre 01.

Através deste sistema hierárquico, mostrado na figura 2-6, o software de gerenciamento pode acessar cada recurso de rede através de seu endereço único. Exemplo: Pode-se acessar o status da porta 7 (Ethernet) do VSAT 1234, NPA 1 (é o único) na partição 1F através do endereço 1F-1234-01-07. [Tridom 02].

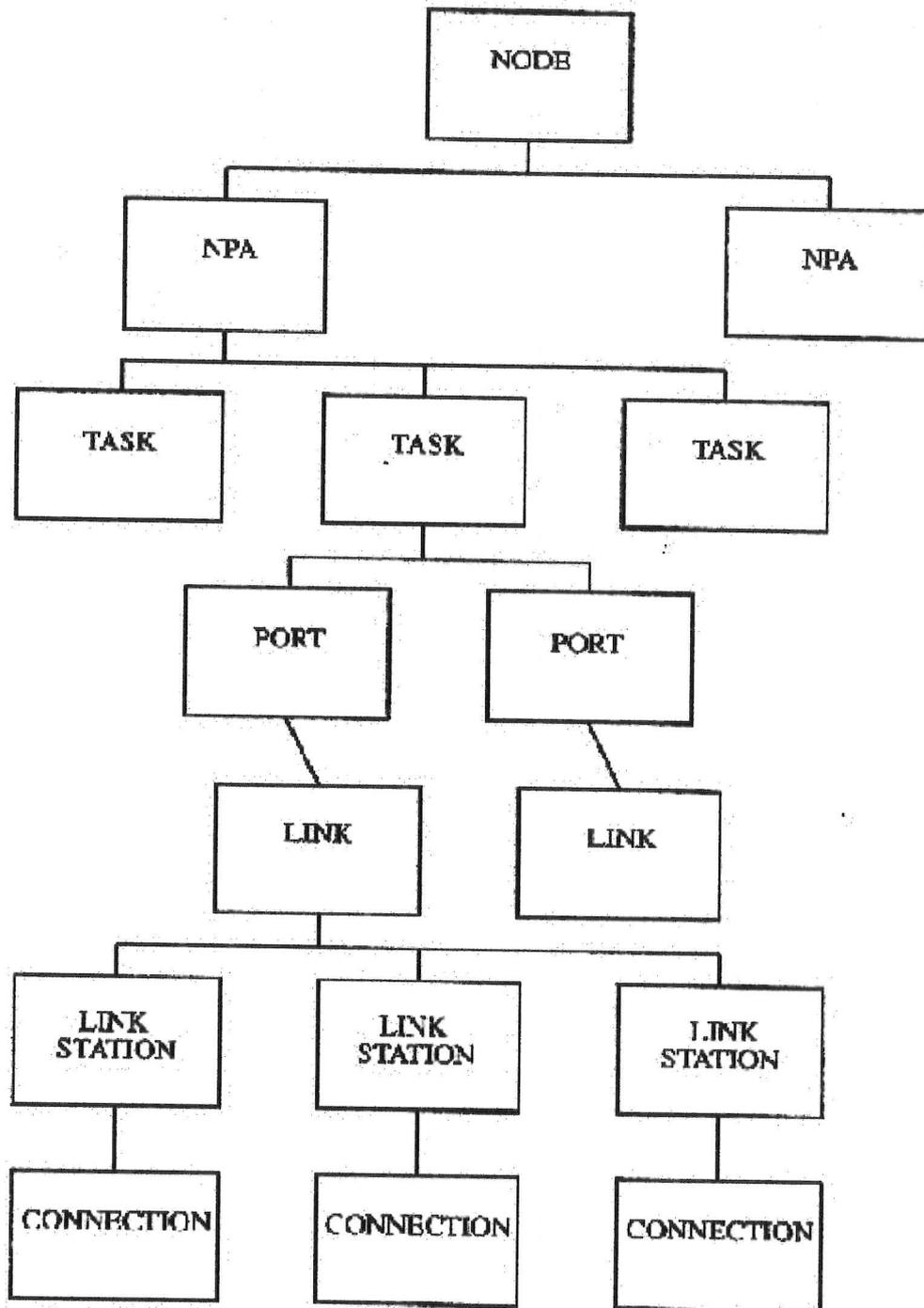


figura 2-6: Hierarquia da rede Clearlink

2.6. Sistema de Controle da Rede Clearlink

Na medida que um rede de comunicação de dados cresce, em número de nós ou na complexidade, o gerenciamento dos recursos da rede torna-se proporcionalmente mais complexo.

O número típico de VSATs presentes em **uma partição** de rede pode atingir a casa dos

milhares em casos de grandes redes corporativas. Para um *switch* (várias partições), o número típico de estações VSAT é na casa dos milhares, chegando a 15 mil nos em *hubs* de grande porte.

Como a maioria dos clientes transporta dados sensíveis para o funcionamento da corporação e precisa das conexões 24 horas por dia, as falhas no sistema devem ser detectadas e corrigidas da maneira mais rápida possível.

A característica de espalhamento geográfico das redes VSAT dificulta a presença imediata de técnicos nos nós da rede com falha. Assim, o *software* de gerenciamento deve possibilitar a execução de diagnósticos, comandos e configurações remotamente, para que se aumente as possibilidades de resolução de problemas a partir do *hub*.

A figura 2-7 traz o sistema de menus do CNCS.



figura 2-7: Sistema de menus do CNCS

2.7. O Software CNCS - Clearlink Network Control System

O software de gerenciamento de rede *Clearlink* é chamado de Sistema de Controle da Rede Clearlink (CNCS - Clearlink Network Control System). O CNCS é executado no NCC e mantém circuitos virtuais para comunicação de gerenciamento como os demais nós da rede. As funções de gerenciamento oferecidas pelo CNCS são: [Tridom 03]

1. Configuração;
2. Status / Monitoração de alarmes ;
3. Estatísticas;
4. Manutenção.

O CNCS permite que ao usuário o controle de uma ou mais redes VSAT a partir de um centro de operações. Este software consiste em módulos (ou tarefas) independentes que executam as funções de gerenciamento em cada ponto de conexão nodal (NPA - Nodal Point of Attachment).

As funções descritas acima estão organizadas em 4 menus de comandos que facilitam os procedimentos operacionais, explicados a seguir: [Tridom 03]

2.7.1. Menu de Configuração

Os comandos deste menu permitem ao operados criar, modificar, listar e apagar os recursos configurados na rede, que são nós (node), processadores de cada nó (NPA), e seus softwares (tasks), portas físicas (ports), enlaces lógicos (links), e estações em cada enlace (link stations). A configuração nas estações VSAT e demais nós da rede é realizada diretamente pelo CNCS, sem necessidade de qualquer intervenção nos equipamentos.

2.7.2. Menu de Status / Monitoração

Este menu permite ao operados acompanhar em tempo real todo o funcionamento da rede, através de comandos de monitoração do estado de todos os nós ou apenas daqueles com problemas de desempenho ou em falha. Dentro deste menú pode-se também listar alarmes gerados, informações sobre a portadora inbound (do satélite para o SS) e localização de uma estação VSAT e sua mudança de portadora inbound dentro da rede.

2.7.3. Menu de Estatísticas

Através deste menu o operador pode acessar relatórios detalhados de todas as medidas efetuadas sobre os recursos de rede. O operador define o recurso a ser monitorado, o tempo total de amostragem e o intervalo entre amostras. Estas funções são úteis para o diagnóstico dos nós de rede. Pode-se por exemplo medir o número de erros de CRC de uma determinada estação a fim de isolar um problema de baixa performance de rede. Pode-se também obter um histograma que indique o espalhamento do número de tentativas de acesso ao meio que um VSAT realiza antes de enviar um dado pelo seu canal. A interpretação deste tipo de gráfico evidencia a eventual sub ou super utilização de um canal.

2.7.4. Menu de Manutenção

Neste menu existem comandos que permitem ao operador controlar a instalação de uma estação VSAT, verificar o conteúdo da memória de um nó, executar a reinicialização, reconfiguração e recarga de um nó (no caso de falha ou modificação de configuração), ativar ou desativar conexões de gerenciamento do CNCS, copiar, apagar ou renomear nós da base de dados, atualizar o mapeamento de portadoras através da adição, modificação ou retirada de canais. Através deste menu também é possível criar operadores e editar seus direitos de acesso.

2.7.5. Operação do CNCS

O software CNCS, executado no computador Stratus (NCC) é composto de processos independentes, divididos em três classes de responsabilidade:

- **Suporte a comunicação.** Processos responsáveis pela conectividade de todos os nós do CNCS, basicamente composto de implementações dos protocolos internos de comunicação;
- **Suporte ao processamento de informação.** Processos responsáveis pelo armazenamento e recuperação de informações relacionadas a rede Clearlink, como alarmes e coleta de dados estatísticos;
- **Suporte a interface de operação.** Processos responsáveis pelo gerenciamento da interface de usuários do CNCS, tratando e executando todos os comandos de operação. [Tridom 04]

O NCC utiliza o software gerenciador de base de dados ORACLE RDBMS - Relational Data Base Management System, que armazena todas as informações relativas ao CNCS. A estrutura interna do Oracle é baseada em SQL - Structured Query Language e, quando executado sob o sistema operacional VOS - Virtual Operational System - dispõe de pré-compiladores para linguagens C, Cobol, Fortran, PL/1 e Pascal. Deste modo, o sistema permite o desenvolvimento, compilação, link, edição de rotinas e aplicativos utilizando estas linguagens, para manipulação das informações armazenadas na base de dados do CNCS. [Tridom 04]

Com vistas no processo de teste que será discutido nos capítulos posteriores é importante ressaltar a diferença entre a configuração de um determinado nó na base de dados do CNCS - a configuração de um VSAT por exemplo - e a configuração residente na memória do nó em questão. Observe a figura 2-8. O CNCS permite que a alteração da configuração de um VSAT seja salva na base de dados sem que isso altere, de fato a configuração no VSAT remoto. Para transferir as configurações para o nó de rede remoto o operador precisa executar um comando específico, independente do comando “salvar alterações para a base de dados”.

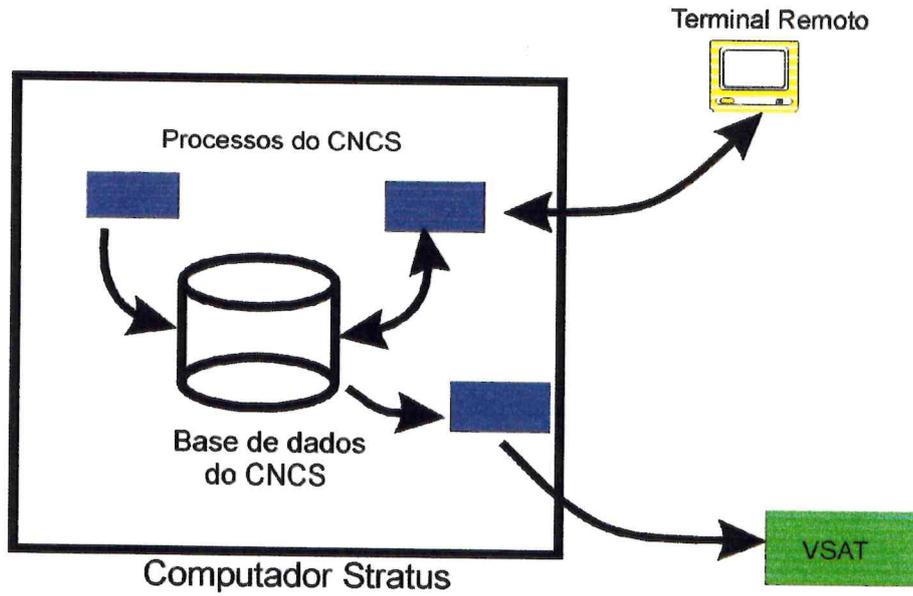


figura 2-8: Base de dados CNCS e nó remoto

2.8. Conclusão

Dado a complexidade da rede em questão, o ponto principal do presente trabalho é definir os limites para um plano de testes do software de gerenciamento da rede, tendo como diretrizes:

- Cobrir a maior quantidade possível de cenários similares aos encontrados na prática;
- Manter o tempo de execução do teste dentro do especificado.

3. Formalização do Problema e Técnicas de Solução: Teste de Regressão do CNCS

3.1. Introdução

O software de controle da rede Clearlink está constantemente sendo aprimorado. Quando um conjunto significativo de alterações é desenvolvido e testado, uma nova versão é lançada no mercado. Em certos casos específicos, um cliente pode receber uma versão de software antes que esta seja formalmente lançada ao mercado para resolver problemas específicos.

No início do projeto, a versão estável era a 9.4.0, e ao final do mesmo, a 9.4.1 já tinha passado para o status “estável” e inclusive testada pelos procedimentos descritos posteriormente nesta monografia. O primeiro uso da rotina de testes final foi na versão 10.0 beta, ainda na fase de testes em laboratório.

Sempre que alterações são realizadas em um software, é preciso testá-lo para saber se o problema inicial foi resolvido e se a alteração não causou outros problemas. Testes deste tipo são conhecidos como testes de regressão (*regression tests*). [Sommerville, 92]

Quando a empresa lança uma nova versão de seu software de gerenciamento e muitos erros são revelados pelo próprio cliente logo no início de sua operação o prejuízo em termos de credibilidade é grande. Uma rede VSAT de 1000 nós custa em torno de US\$ 15 milhões, e em geral a data de início de operação da rede é um ponto crítico.

O custo operacional para contornar problemas quando o software já foi distribuído também é alto, em especial para clientes internacionais.

O objetivo deste capítulo é discutir possíveis soluções para estes problemas.

3.2. Verificação e Validação de Software

Verificação e validação (V&V) são termos genéricos para procedimentos de teste que vão avaliar se o software atende aos requisitos estabelecidos nas primeiras etapas do desenvolvimento do software. A verificação e validação de um sistema de software é um processo contínuo, que está presente em cada estágio do desenvolvimento do software. Os objetivos deste processo são:

1. Descobrir defeitos no sistema
2. Verificar se o sistema é utilizável em determinadas situações práticas
[Sommerville, 92]

A diferença entre **validação** e **verificação** é resumida por Boehm:

- Validação: Estamos construindo o produto certo?
- Verificação: Estamos construindo de maneira certa o produto?
[Boehm, 1979]

Verificação vai responder se o software atende as especificações de projeto. A validação

indica se o software atende as expectativas do usuário (ou cliente).

Para alcançar os objetivos do processo de V & V, pode-se usar técnicas de teste estáticas ou dinâmicas. A **verificação estática** é baseada na análise e exame do código fonte do programa. A execução não é necessária. As análises matemáticas ou lógicas do código fonte entram nessa classe de testes. A **verificação dinâmica**, por sua vez, requer a execução do programa e a montagem de cenários típicos para

Alguns autores sugerem que as técnicas estáticas podem substituir por completo a verificação dinâmica. Outros, como [Sommerville, 92], afirmam que isto não faz sentido, pois as técnicas estáticas se limitam a verificar a correspondência entre especificação e o programa final, não podendo demonstrar que o programa seja útil operacionalmente.

A abordagem de Sommerville foi utilizada neste trabalho devido à característica do problema. Dada o número de protocolos de acesso, e do ambiente de desenvolvimento, a melhor maneira de evitar que o usuário descubra os problemas é testando o mesmo em uma série de cenários típicos que simulem o seu uso normal. Os procedimentos de teste do CNCS apontados como solução neste trabalho são puramente dinâmicos.

Testes de programas podem apontar erros no programa, mas não podem mostrar que o programa é livre de erros. Mesmo após os testes mais exaustivos qualquer programa ainda pode conter erros não detectados.

Um teste de defeitos satisfatório é aquele que, de fato, revela defeitos no programa [Myers, 1979]. Na metodologia veremos, por exemplo, que o teste de telas foi colocado em segundo plano, já que não revelou nenhum defeito durante duas execuções.

3.3. Metodologia de teste anterior ao projeto

A cada versão ou revisão nova do CNCS, instalava-se o software correspondente em algumas partições do laboratório de redes da empresa. Assim, a nova versão ou revisão ficava disponível para uso de toda a engenharia. Como o laboratório estava sempre em uso para testes e desenvolvimentos diversos, a versão nova do CNCS seria supostamente testada nestas atividades.

Estudando os a base de dados de problemas reportados (FOPs), observou-se que poucos ou nenhum problema era revelado neste tipo de teste. O motivo é que os usuários (da engenharia) não tinham como objetivo explorar diferentes configurações ou tocar em diversas partes do software e sim executar os testes que lhes interessassem. Assim, uma parte muito pequena do software eram testada, fazendo com que os defeitos passassem despercebidos para a versão comercial.

Outro tipo de teste que ocorria ficava por conta dos responsáveis pelas alterações ou adições ao software. Se a equipe A, por exemplo, acrescentasse uma nova funcionalidade no X.25, esta mesma equipe teria de fazer o teste de regressão para o protocolo X.25 antes que a nova versão fosse lançada no mercado. Como o teste de regressão é trabalhoso e consome muito tempo, nem todas as equipes testavam adequadamente o que deveriam.

3.4. Definição do Escopo do teste

Conforme exposto anteriormente, o ideal seria se testar **exaustivamente** a nova versão ou revisão, a fim de evitar que erros fossem encontrados após a distribuição do software.

Contudo, um teste completo, que cubra todas as possibilidades é inviável na prática. O motivo é o tamanho e a complexidade do CNCS, que possui em torno 150 de telas. Em dezembro de 97 o produto Clearlink suportava 30 protocolos de acesso. Cada protocolo de acesso ou protocolo da rede possui telas de configuração para os níveis (de um a 6) da hierarquia da rede Clearlink descrita anteriormente.

Em uma estimativa inicial, para testar somente as configurações relativas aos protocolos de acesso e internos seria necessário testar mais de 120 telas, em rotinas de adicionar, modificar, transferir para a rede e apagar da base de dados do CNCS. Em cada passo teria que ser verificado se os dados em questão foram de fato alterados.

Cada tela possui em média 10 campos, numéricos, de texto ou múltipla escolha, totalizando $120 \text{ telas} \times 10 \text{ campos} \times 4 \text{ operações} = 4800 \text{ iterações}$. Considerando que cada iteração leva em média 5 minutos, uma estimativa razoável, seria preciso 400 horas (ou 16 dias e meio trabalhando 24 horas por dia) para testar somente a parte de configuração do CNCS. De posse desses dados, concluiu-se que o principal seria definir o escopo do teste, de modo que o teste fosse o mais abrangente, próximo da realidade do usuário e que consumisse o mínimo de horas de um mínimo de pessoas na equipe de testes trabalhando com um mínimo de equipamentos.

Em termos de qualidade de software, um sistema pode medido basicamente em quatro dimensões: eficiência, manutenção, exatidão e confiabilidade [Gillies, 1997]. No presente trabalho a dimensão de qualidade em foco é a exatidão. Deseja-se saber se o software responde de maneira adequada aos comandos do usuário. Mais especificamente, deseja-se saber se as configurações chegam aos devidos nós da rede passando pela base de dados.

Também não se faz necessário testar todas as possibilidades, como se faz em softwares de uso crítico, para naves espaciais, sistemas de tracking de satélites ou de uso militar. Também não é o caso de testar todas as combinações possíveis em todos os campos.

Há casos em que mesmo que o defeito exista, este não vai ser consertado, pois não se trata de uma situação de uso cotidiano. Como o usuário do CNCS na prática é um operador com um bom nível de treinamento (ou não há razão para que este trabalhe com o software), pode-se descartar uma série de possibilidades que teríamos de levar em consideração no caso de um usuário comum.

Um exemplo para isso é uma das falhas encontradas no CNCS durante os testes preliminares. Na configuração do protocolo TDMA que faz o link entre os VSATs e o HUB, existe um campo que configura o tamanho da janela entre dois pulsos de sincronização da rede (superframe time, ou SFTM interval). A unidade neste campo é décimos de segundo, e um valor típico encontrado neste campo é 100, que indica uma janela de 10 segundos.

Este campo permite a entrada do valor 9000000, ou seja, $9e5$ segundos, ou 10.4 dias, um valor totalmente absurdo na prática, nenhum operador digitaria este valor. Contudo, se entrarmos com este valor de janela de tempo, ao configurar o link equivalente o CNCS trava. Problemas

deste tipo, que pouco provavelmente serão tocados ou atingirão algum cliente são despriorizados pela engenharia, que se ocupa com as emergências e com a próxima geração de software.

Por isso a importância de uma vivência prática com o ambiente de operações de rede VSAT, para eliminar procedimentos que não seriam úteis para identificar problemas que potencialmente seriam identificados pelo cliente.

Em vários protocolos, se não em todos, há parâmetros que nunca são alterados em situações práticas. Em geral o cliente altera um conjunto limitado de parâmetros para cada protocolo. Um exemplo é o parâmetro “addressing” da porta LAPB, que determina se a porta usa ou ignora endereçamentos. Em conversas com especialistas na área concluiu-se que este parâmetro muito raramente é alterado.

Há também os parâmetros que são alterados somente pela engenharia Tridom, como na configuração da porta BLAPB, o Xmit TMO Threshold. Estes parâmetros são alterados raríssimas vezes, não sendo alvo principal de testes.

3.5. Metodologia para encontrar a solução

O problema foi exposto como “Precisamos de um teste para o CNCS. O número de defeitos encontrados em versões novas está acima do aceitável e isso além de prejudicar a qualidade do produto aumenta os custos da empresa, principalmente em clientes internacionais”.

O presente trabalho começou, de fato, com a definição do problema, que não era necessariamente criar um teste, mas *avaliar o processo anterior ao lançamento do software no mercado* e propor soluções. Que partes do CNCS estavam sendo tocadas com a metodologia atual? Que tipo de teste seria necessário? O que se deseja testar? O que é possível testar? Com que recursos? Em quanto tempo?

A raiz do problema é a atual metodologia está deixando passar defeitos que poderiam ser detectados em laboratório para a versão comercial.

Para chegar ao delineamento de uma rotina de testes partindo da definição vaga e informal do problema, passou-se pelas seguintes etapas:

- ✓ **Etapa 1:** Embasamento teórico em redes VSAT e protocolos.
Recursos: Treinamento formal na AT&T Tridom e bibliografia especializada
- ✓ **Etapa 2:** Estudo e definição formal do problema
Recursos: Base de dados de problemas identificados nas versões comerciais (FOPs), Entrevistas com engenheiros envolvidos nas modificações do software (para levantamento de protocolos utilizados, sugestões e procedimentos que cada um utilizava para testar a sua seção), estágio em um centro de operações de redes, para analisar a utilização prática dos sistemas em questão.
- ✓ **Etapa 3:** Elaboração de um roteiro de teste piloto.
Recursos: Dados adquiridos na etapa 1 e 2.
- ✓ **Etapa 4:** Aplicação do teste e revisão do roteiro. Elaboração da documentação do plano de testes nos padrões da empresa.

Recursos: Equipamento para configurar uma rede VSAT, que inclui um SS e um HI instalados em uma partição isolada da rede, feedback dos orientadores do teste, modelo de procedimentos de teste

3.6. Conclusões

Conforme demonstrado, a questão principal é definir o escopo do teste, dado que um teste exaustivo seria impossível. Para definir as margens do teste é preciso conhecer bem o software, a arquitetura da rede, as configurações dos protocolos disponíveis, os procedimentos e configurações típicas utilizadas na prática e os processos de teste realizados pelos programadores.

O presente trabalho tem como objetivo a **verificação** dos softwares em questão, já que deseja-se identificar defeitos no software pronto.

A validação **dinâmica** é necessária nesse teste. A verificação estática é responsabilidade das equipes que desenvolvem ou modificam os protocolos de acesso e o núcleo do CNCS.

Deseja-se chegar a um procedimento de teste de regressão do software de gerenciamento da rede que possa ser executado em uma semana por dois operadores. O teste deve atingir os protocolos mais comuns, usados no mínimo por 80% dos clientes. Espera-se detectar em laboratório cerca de 20% do total de problemas relatados durante os primeiros 6 meses em que o software esteja no mercado. Com a atual metodologia de teste, menos de 10% dos problemas são detectados em laboratório.

Apesar de o número 20% parecer pequeno, esta é uma porção muito significativa, pois em torno de 50% dos problemas relatados são muito específicos, dependentes da aplicação da rede, sendo inviável a reprodução em laboratório. Os restantes 30% poderiam ser detectados por um teste exaustivo

4. Elaboração e Implementação do Plano de Testes

4.1. Introdução

A elaboração do plano de teste foi um processo iterativo, repetindo as etapas 3 e 4 expostas na seção anterior. A avaliação do teste inclui:

1. Medir o tempo de execução, cobertura (só testes de telas/ só interações com a base de dados / incluir a configuração até o download na rede). O teste funciona? Ou em outras palavras: achamos bugs? Quais os bugs da base de dados de FOPs seriam efetivamente encontrados com o teste que testamos?

2. Avaliação por parte dos analistas de sistemas envolvidos com a manutenção do CNCS, diretor do departamento de verificação e validação (SVE) e orientador. O teste serve? Sobreposição algum teste realizado normalmente?

4.2. Cronograma utilizado:

Atividade	Duração	Início	Término
Estudo e definição formal do Problema	2	1/ago/97	15/ago/97
Estudo da arquitetura de rede VSAT	4	1/ago/97	29/ago/97
Estudo do CNCS, funcionamento geral, configuração básica de rede.	4	18/ago/97	15/set/97
Entrevistas com Analistas de sistemas	7	15/ago/97	22/ago/97
Análises dos FOPs	7	15/ago/97	22/ago/97
Estágio em um Centro de Operações de Rede de VSAT - Hub Atlanta, GA	30	1/out/97	31/out/97
Elaboração da Proposta Inicial de Roteiro para teste do CNCS	8	25/ago/97	24/out/97
Testes preliminares do CNCS ver. 9.4.1: Salvar para base de dados apenas.	15	25/out/97	5/nov/97
Estudo de viabilidade de automação dos testes	6	13/out/97	21/nov/97
Reavaliação da rotina de teste proposta e elaboração do teste final	8	24/nov/97	2/dez/97
Teste piloto ("dry run") com a versão 10.0 Base de dados apenas	15	2/dez/97	16/dez/97
Avaliação dos resultados, revisão da rotina de teste.	3	16/dez/97	19/dez/97
Teste final, versão 10.0 (antes do lançamento), na base de dados e rede montada em laboratório	14	6/jan/98	10/jan/98

Cursos Programados			
Curso: <i>AT&T Tridom Network Architecture Overview / Data Flow</i>	3 dias	27/jan/97	29/jan/97
Curso: <i>VSAT Installation and Maintenance</i>	1	18/ago/97	22/ago/97
Curso: <i>CNCS Private HUB Program</i>	2	8/set/97	19/set/97
Curso: <i>VSAT Network Design Considerations</i>	3 dias	22/set/97	24/set/97
Curso: <i>Aegis Private HUB Program</i>	1	27/out/97	31/out/97

4.3. Coleta de dados para desenhar a solução

4.3.1. Análises dos relatórios de defeito (bug reports ou FOPs)

Sempre que um operador da rede, programador engenheiro ou usuário (através de um dos três primeiros) detecta um problema no software da rede Clearlink, isto é formalizado através de um relatório de defeito. Estes relatórios alimentam a base de dados dos FOPs, que fica disponível na Intranet para que os responsáveis trabalhem nos problemas de acordo com as prioridades.

A seguir temos dois exemplos de FOPs, retirados da base de dados Tridom. O primeiro FOP ilustra um tipo de problema que seria detectado em um teste simples de configuração que envolvesse download da configuração para a rede. O bug report afirma que as configurações do protocolo ALAPB não chegam corretamente ao nó de rede desejado. O autor do FOP acrescenta os resultados da de comandos ODB, que "lê" as configurações recebidas pelo nó remoto. Observou-se que cerca de 20% dos FOPs eram desta natureza.

Já o segundo FOP apresentado apresenta um problema de compatibilidade entre CNCS e Aegis, na seção de estatísticas, que não seria detectado por um teste de configuração. Contudo, erros assim são menos comuns e de menos impacto para o usuário, diga-se de passagem que o problema existia a dois anos.

---- THE FORMATTED ISSUE FOLLOWS ----

Issue: I...-450
Version: 1.2
Last changed: 1997/08/01,10:38:43

PROMOTION STATUS

Promoted to: Submitted On: 1997/06/26,16:06:59 By: qwan

CURRENT ATTRIBUTES

Title: CNCS v9.4.1 and v10.0 - ALAPB Link configuration
Originator: Qin Wan
Issue Type: Problem
Customers Affected: All customers who need ALAPB Link
Customer Impact: Severity-1
Release Type: Production

```

Platform:          CNCS/Stratus
Responsibility:    Netmgmt
Group:            ---
Priority:          High
Team_Lead:
Assigned To:
Est_Manhours:
Due_By:
Problem Cause:    ---

```

STATEMENT OF PROBLEM

Statement of problem...

When ALAPB Link configuration is sent down to network on both CNCS v9.4.1 and CNCS v10.0, if "Operation Mode" is configured as Passive, the configurations of "Sequence Numbers" and "DM Responses", which are both represented by the bits of lvl2Flags (bit 3 - 0x08 and bit 1 - 0x02 respectively) in ALAPB_LINK_CONFIG structure, can be sent down correctly. However, if "Operation Mode" is configured as Active, the "Sequence Numbers" and "DM Responses" are always sent down with default values which are "MOD_8" and "Enable" respectively.

Here is the SMSE message when configuring "Operation Mode"(bit 0 - 0x01 of lvl2Flags) as PASSIVE, "Sequence Numbers" as MOD_128, "DM Responses" as Disable, "Poll Supervisory" (bit 2 - 0x04 of lvl2Flags) as Enable, and "Preferred Acknowledgement"(bit 4 - 0x10 of lvl2Flags) as I-Frame:

```

-----
0xcfc4e: REQUEST (NCSE):rsp req. Cat: CONFIG Code: MODIFY_CONFIGURATION
Resource[P29v0s0F0]: LINK, NPA #1, Port #2, Protcl #29, ID 0x0000
User data (74 bytes):
01 2f 00 1d 01 0e 00 01 00 15 23 28 07 05 03 e8
00 c8 01 05 00 1e 02 01 00 00 00 00 00 00 00 00
      ^^ ^^ (binary: 0000 0000 0001 1110, which is correct)
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
-----

```

The following is the SMSE message when configuring "Operation Mode" as ACTIVE and all other parameters the same as above:

```

-----
0xd29c6: REQUEST (NCSE):rsp req. Cat: CONFIG Code: MODIFY_CONFIGURATION
Resource[P29v0s0F0]: LINK, NPA #1, Port #2, Protcl #29, ID 0x0000
User data (74 bytes):
01 2f 00 1d 01 0e 00 01 00 15 23 28 07 05 03 e8
00 c8 01 05 00 15 02 01 00 00 00 00 00 00 00 00
      ^^ ^^ (binary: 0000 0000 0001 0101, it's not correct.
            It should be 0000 0000 0001 1111, which is 001f hex)
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
-----

```

```

Loadfile:
Exceptions, Alarm logs, etc...
Notes:

```

STATEMENT OF SOLUTION

Actions taken...

Modified udlc_cmds.c to fix this problem. Sent an email to the originator of this fop to test the fix.

```

Work-around:
ECR Number:

```

```

Component(s) Modified:
Files Modified:
Functions Modified:

```

New Loadfile:

Notes:

Exemplo 2:

---- THE FORMATTED ISSUE FOLLOWS ----

Issue: I...-224
Version: 1.5
Last changed: 1997/06/18,14:44:01

PROMOTION STATUS

Promoted to: Submitted On: 1995/07/11,10:12:23 By: srd

CURRENT ATTRIBUTES

Title: CNCS-9.1 Statistics probs
Originator: Scott Dennis
Issue Type: Problem
Customers Affected:
Customer_Impact: Severity-2
Release Type: Production
Platform: CNCS/Stratus
Responsibility: Netmgmt
Group: ---
Priority: Low
Team_Lead: Keith Thorne
Assigned_To:
Est_Manhours:
Due_By:
Problem Cause: Reqts

STATEMENT OF PROBLEM

Statement of problem...

The new 9.1 CNCS Statistics has some nasty things about it (of course). When the statistics are started, if the node address is entered as 01-0002, then Aegis doesn't find the DB entries and 01-0002 must be entered in the review stats screen. If the node address is entered as 1-0002, then everything works. The problem is - the screen always displays 01-0002 which of course confuses everyone. Also, is anyone ever going to do anything about FOP 142. The FOP was entered this year but Eng. has known about it for about two years. It is very irritating.

Loadfile:

Exceptions, Alarm logs, etc...

Notes:

STATEMENT OF SOLUTION

Actions taken...

Work-around:

New Loadfile:

ECR Number:

Notes:

A base de dados dos FOPs foi muito útil para determinar que tipos de problema poderiam ser detectados por qual tipo de teste. Acompanhamos os FOPs gerados por todas as fontes por dois meses, procurando determinar com que frequência que cada tipo de problema aparecia. Daí saiu a conclusão que a etapa do teste de telas e menus era inútil, conforme será discutido posteriormente.

A idéia da análise dos FOPs foi determinar como poderíamos planejar um teste dentro dos limites planejados - 10 dias - e que potencialmente detectasse o maior percentual de erros possível dentro da categoria de erros que trouxesse mais impacto ao cliente.

4.3.2. Entrevistas com desenvolvedores de software

Conversamos com boa parte da equipe de desenvolvimento de software da AT&T Tridom, mais intensamente durante a terceira semana de agosto de 97. Procuramos os especialistas nos produtos mais populares como Roteador IP, SDLC e X.25. Procuramos também os responsáveis pelos desenvolvimentos mais recentes como Frame Relay.

Os objetivos deste procedimento foram:

1. Determinar que protocolos eram mais utilizados na prática
2. Coletar sugestões sobre procedimentos de teste
3. Determinar que partes do software eram testadas no processo atual e como.

Para determinar os recursos de rede mais frequentemente utilizados, elaboramos uma tabela com todos os recursos de todos os níveis hierárquicos da rede e solicitamos que os programadores indicassem quais os produtos tinham base instalada considerável e que deveriam ser testados. Para isso usamos a tabela mostrada nas tabelas abaixo (figura 4-1)

Tasks							
	IP		TP4		MAILMAN		CVCC
	TLCP		LAPB		DSP		KDIR
	KPAD		MPAD		INET		X.25 L3
	X.25 ATP		CONN SRV		ASBS		TLCC
	TLCS		MUX		ETHERNET		SLIP
	751 BSC		3780 BSC		VDLC		BLAPB
	TRLC		TRMR		TNET		L2PVC
	GDS		VCS		HYDRA		P1024
	CREDIT		TDLS		P1024C PPAD		TOPO
	LAP- PIPE		FRLC		GDS2		X.75

Ports			
BSC		SDLC	ASYNC
LAPB		TLCP	CSO LAPB
CSO PLUS		L2PVC	X.25 L3
ASBS		MAP II CP	MAP II CS
MAP II MUX		HAYES MPAD	SCADA MPAD
SLIP		ETHERNET	751 BSC
3780 BSC		VDLC	BLAPB
TRLC		TNET	HYDRA
P1024		TDLS	P1024C PPAD
LAP-PIPE		FRLC	X.75

Links			
LAPB		TLCS	P1024
SDLC		MUX	TDLS
BSC		751 BSC	T1024C
L2PVC		3780 BSC	PPAD
X.25 L3		VDLC	
X.75		BLAPB	
LAP-PIPE		TRLC	
TLCC		HYDRA	

Link Stations				
LAPB-PVC		SDLC	BSC	L2PVC
X.25 L3		KPAD	751 BSC	3780 BSC
VDLC		TRLC	TPAD	P1024
CREDIT		P1024C	PPAD	LAP-PIPE
MPAD		CLPAD	MODBUS	FRLC
XPAD				

figura 4-1: Tabelas com os todos os recursos da rede Clearlink

Numa etapa posterior, passamos as configurações dos VSATs utilizados no plano de teste piloto para os programadores. O objetivo foi confirmar se as configurações propostas no teste eram coerentes com as utilizadas na prática.

4.3.3. Verificação de configuração através de comandos ODBC

Um dos testes realizados por algumas equipes de desenvolvimento consiste em verificar através das portas ODBs se os parâmetros de configuração armazenados na base de dados do CNCS são corretamente transferidos para o nó da rede em questão (um VSAT, por exemplo).

A porta ODB é uma porta serial para auxílio no diagnóstico dos nós da rede Clearlink, ou seja, VSATs, Host Interfaces e Switching Systems. Para efetuar o diagnóstico é preciso conectar-se a esta porta (através de telnet, por exemplo) e executar os comandos definidos para cada

recurso de rede. A saída de um comando ODB é um bloco de números em hexadecimal que decodificados contém o valor dos campos de configuração daquele recurso. O resultado típico de um comando ODBC é mostrado abaixo (figura 4-2):

```
0xcfc4e: REQUEST (NCSE):rsp req. Cat: CONFIG Code: MODIFY_CONFIGURATION
Resource[P29v0s0F0]: LINK, NPA #1, Port #2, Protcl #30, ID 0x0000
User data (74 bytes):
01 2f 00 1d 01 0e 00 01 00 15 23 28 07 05 03 e8
00 c8 01 05 00 1e 02 01 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

figura 4-2: Resultados de um comando ODB

O resultado acima se refere a um comando de configuração da porta 2, configurada para o protocolo número 29 (BLAPB), no NPA 9 de um determinado nó de rede. A tela do CNCS correspondente é mostrada a seguir:

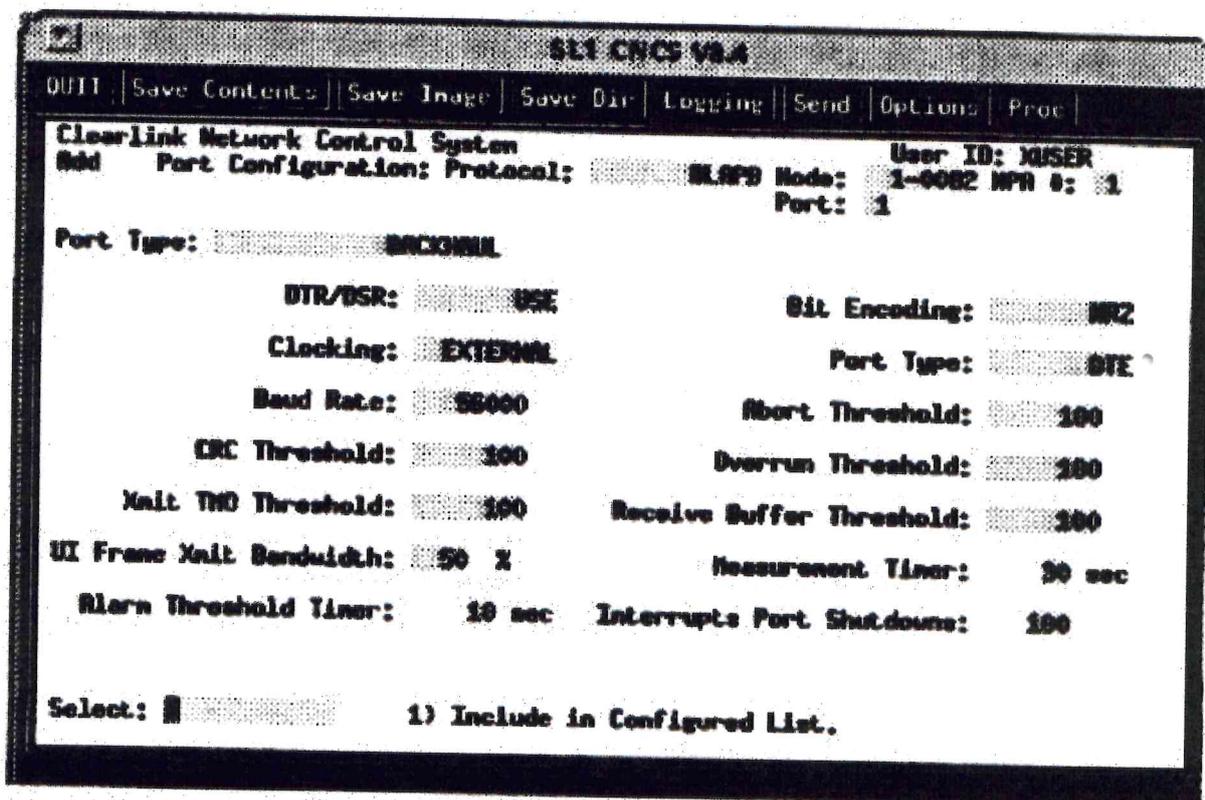


figura 4-3: Tela de configuração de uma porta LAPB

Este procedimento é demorado e exige conhecimento do código em C envolvido para encontrar a equivalência entre os códigos em hexadecimal e os parâmetros de cada recurso de rede. Entretanto este é um teste completo que, quando bem executado, verificaria por completo a configuração do recurso de rede testado, pois consiste na verificação do processo desde a inserção na base de dados até o download para os nós de rede (VSATs, HIs, ou SSs).

Em princípio a rotina de testes que realizaríamos deveria incluir a verificação pela porta ODB mas, dada a complexidade do processo, concluiu-se que seria melhor garantir que os próprios programadores realizassem este tipo de verificação, que seria muito mais fácil e rápida já

que os programadores estariam mais familiares com o código e as interpretações dos resultados dos comandos ODB.

O processo de verificação de configuração pela porta ODB pode ser resumido em três passos basicamente:

1. Inserir uma configuração na base de dados do CNCS (adicionar a porta 2, configurada para X.25, por exemplo, no VSAT de endereço 1-8015)
2. Transferir a configuração para o nó de rede em questão. Exemplo: O VSAT 1-8015 que proporcione acesso a X.25 pela porta 2)
3. Conferir se as configurações que o VSAT recebeu (no caso, o 1-8015), através da porta ODB.

Observamos que nem todas as equipes faziam o teste usando comandos ODB. Alguns programadores desenvolviam um conjunto de comandos ODB para os protocolos que trabalhavam mais frequentemente. Com estes comandos era possível montar scripts que possibilitavam a comparação rápida entre a configuração desejada (na base de dados) e a configuração real (em um determinado nó da rede). Comandos ODB facilitam o debug do software, porém nem todos os programadores fazem uso desta ferramenta.

4.3.4. Estudo da automação dos testes

A porta ODB seria a saída para automação dos testes. Seria possível escrever scripts que rodassem no computador Stratus, que através de comandos SQL iria criar as configurações desejadas na base de dados, transferir para os nós da rede, e ao final, a saída dos comandos ODBC seria decodificada e comparada com as configurações originais na base de dados.

O script poderia conter as rotinas de adição, modificação e exclusão de cada recurso de rede, podendo-se inclusive planejar um teste que cobrisse todas as telas e todos os campos possíveis.

O problema é que o conjunto de comandos ODB não é padrão entre os recursos de rede e além disso, a decodificação da saída destes comandos é fortemente dependente do tipo de recurso que se está testando.

Em alguns casos, os programadores desenvolveram comandos que já vem com a saída decodificada, o ideal para utilização em um script, mas na maioria dos casos comandos como este não existem.

O problema da decodificação é visível quando compararmos a tela de configuração do CNCS e a saída do comando ODBC apresentados anteriormente.

4.3.5. Estágio em um Centro de Operações da Rede - Hub Atlanta, GA.

Centro de Operações da Rede (NOC ou Network Operations Center) é o local de onde se faz as configurações, monitoramento e manutenção dos nós da rede. O monitoramento pode ser proativo ou reativo.

No caso reativo esperava-se que alguém do de um ponto VSAT informe que há um

problema em seu site para então dar início a manutenção. No caso reativo, monitora-se os pontos de interesse ou toda a rede através do CNCS e/ou Aegis para detectar eventuais problemas independentemente do usuário.

O NOC realiza a instalação lógica dos VSATs na rede após a instalação física da antena, cabos e equipamento no local remoto. Isto permite que o VSAT se comunique com os demais elementos da rede. A instalação lógica consiste em inserir a configuração apropriada na base de dados e transferir esta configuração para o VSAT. Outros testes são realizados a partir do NOC, como a medição da potência e polarização da portadora recebida do VSAT. Estes testes e ajustes vão assegurar o correto funcionamento da estação VSAT.

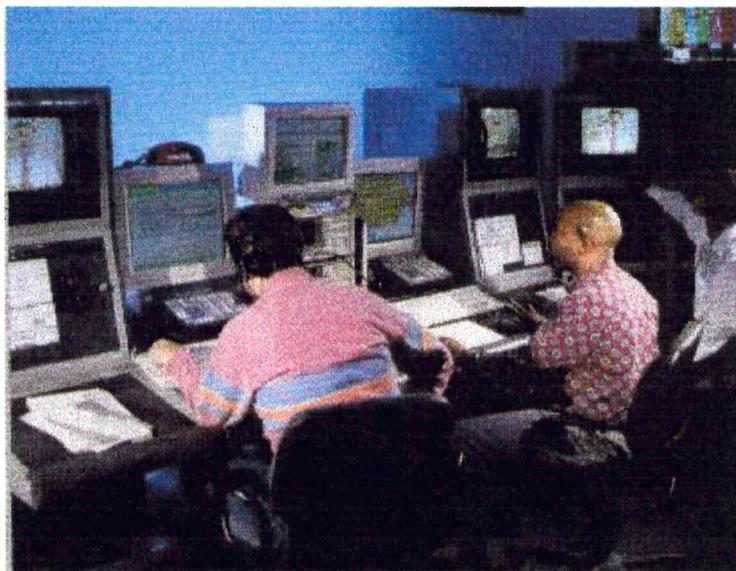


figura 4-4: Cenário típico de uma divisão de um NOC

O NOC representa, afinal, o grupo de usuários dos softwares de gerenciamento da rede VSAT - CNCS e Aegis. Daí a importância de se conhecer os procedimentos utilizados neste ambiente para elaborar um teste. A equipe do NOC da Tridom que detecta formalmente os problemas da rede relatados pelo cliente final, o usuário do VSAT.

Existem dois tipos de HUB, privado e compartilhado. Clientes com redes VSAT grandes (mais de 1000 VSATs) em geral utilizam HUBs privados. Em certos casos estes clientes grandes possuem um NOC para controle de sua rede. Um exemplo disto é o banco Bradesco.

O Hub compartilhado controla a rede de vários clientes em um NOC. Este é o caso de alguns Hubs da Embratel e do NOC de Marietta, GA, onde realizamos os estágio. A experiência em um HUB compartilhado foi ideal, pois foi possível observar as diferentes aplicações para a rede VSAT e ter uma melhor noção de como o teste deveria ser planejado para simular as situações encontradas em um HUB como este.

4.4. Conclusões relativas à elaboração do teste e implementação

Em termos gerais, pode-se afirmar que é possível melhorar o procedimento de testes atuais

para diminuir consideravelmente o número de erros de software encontrados em campo. Note-se a diferença entre aumentar o número de defeitos encontrados em laboratório e diminuir o número de defeitos encontrados em campo.

Quanto mais exaustivo o teste em laboratório, naturalmente mais defeitos serão encontrados. Contudo, o esforço do teste é pouco aproveitável se os defeitos encontrados não forem relacionados com situações de uso prático em campo.

4.4.1. Sobre as entrevistas com os programadores

As principais conclusões tiradas desta etapa:

1. É necessário que os responsáveis executem o teste de regressão nos novos desenvolvimentos ou após a manutenção dos módulos do CNCS. A gerência das equipes garantir estes testes.
2. Para automação dos testes futuros é preciso que se desenvolva comandos ODB para acesso e decodificação das configurações dos nós de rede (VSATs, HIs ou SSs).
3. É necessário que se estabeleça uma documentação dos comandos ODB e dos testes realizados em laboratório, que era praticamente inexistente no início do projeto.
4. Chegamos a um conjunto de protocolos mais utilizados na prática. O plano de testes foi planejado para atingir estes protocolos.
5. Definiu-se também os parâmetros típicos de configuração de cada recurso de rede. (Exemplo: as configurações dos parâmetros de uma porta X.25)
6. Seria necessário fazer o teste completo, até a etapa final onde se transfere as configurações para os nós de rede. As interações com a base de dados representavam uma parte mínima dos problemas gerados em campo. Este fato foi confirmado nas análises dos FOPs.

Com relação à automação dos testes, concluiu-se que isto depende da padronização dos comandos ODB por parte dos desenvolvedores de software. Antes disso fica impossível a automação completa. A parte do teste que envolve a configuração dos nós pode ser automatizada através de scripts para acesso à base de dados. Contudo, a parte de configuração no nível da base de dados é a que consome menos tempo no processo.

4.4.2. Sobre o estágio no NOC de Atlanta, GA.

As principais conclusões que direcionaram a elaboração do plano de testes foram:

1. Não é preciso testar os valores mínimos e máximos de cada campo, pois isso não é estressado em situações práticas.
2. As rotinas de instalação são bem mais simples do que as propostas nos testes preliminares. Os operadores usam templates para instalar VSATs, o seja, existe um conjunto determinado de configurações que são copiadas na base de dados.

Para instalar logicamente um VSAT para uma aplicação de voz, por exemplo, o operador copiaria a configuração de um template de VSAT que contém uma configuração genérica. Na maioria dos casos pequenas alterações são feitas no padrão para adaptar a configuração à necessidade do usuário final.

Contudo, optamos por construir os nós a partir do zero, adicionando todos os recursos de rede para cada nó (NPAs, Ports, Links, etc.). O motivo para manter a metodologia diferente da utilizada no NOC é que este procedimento não acrescentaria muito tempo ao teste, além de que o teste de telas e menus seria coberto desta forma.

3. As configurações mais típicas foram identificadas e, diga-se de passagem, estas são muito menos complexas do que se esperava. Em linhas gerais, as configurações mais utilizadas envolviam SDLC, IP router e X.25. Pouco menos frequentemente encontrava-se VDLC, um protocolo que em geral acompanhava uma das três configurações acima.

4. Outra conclusão importante, foi sobre o teste do Aegis, que foi passado para outra equipe de trabalho já envolvida no processo de validação destes softwares para o ano 2000.

Conforme visto anteriormente, as funções de um software de gerenciamento de rede incluem:

- Coleção de estatísticas;
- Diagnósticos;
- Configurações

Vimos nos capítulos iniciais deste trabalho que o Aegis não executa praticamente nenhuma função de configuração, e que o problema formalizado tinha como causa principal os defeitos na parte de configuração da rede, que são executadas pelo CNCS. Por esta razão, concentramos esforços na elaboração de um teste para o CNCS.

A parte crítica da rede VSAT é a configuração. No momento em que o operador do NOC está tentando solucionar um problema em campo, se os parâmetros que se vê na tela (que vêm da base de dados) não forem condizentes com os que estão no VSAT o diagnóstico fica extremamente difícil.

5. Resultados

Após a etapa de estudo do problema, da elaboração e aplicação de um teste preliminar, chegamos a um plano de testes definitivo.

5.1. Descrição do Plano de Testes Proposto

O teste proposto envolve três operações básicas da base de dados do CNCS, adicionar (add), modificar (modify) e apagar (delete). Após o passo “modificar” transferíamos a configuração para a rede e simulava-se algum tráfego a fim de verificar se os nós foram configurados.

Uma partição do laboratório com dez VSATs, dois HIs e um SS foi montada para a realização do teste. O teste começaria na instalação do CNCS no computador Stratus. A partir daí seriam construídas as configurações do SS e de cada HI e VSAT definidos pelo plano de teste.

As configurações que cobrem os 80% das aplicações dos clientes estão incluídas em anexo. Trocamos algumas placas do SS e do HI para que fossem compatíveis com as configurações e para que a rede funcionasse ao final. As configurações definidas seguem principalmente os resultados das entrevistas com os programadores.

Durante a sua inicialização o VSAT recebe do NCC, via satélite, seu sistema operacional e o software adicional necessário para a operação dos protocolos para os quais foi configurado. O conjunto de arquivos recebidos pelo VSAT é empacotado em um único arquivo, ou loadfile.

Por limitação de tamanho o loadfile não pode conter a funcionalidade de todos os protocolos suportados pela rede Clearlink. Por um número limitado de combinações de protocolos estão disponíveis. Para o teste elaborado, escolhemos configurações que fossem compatíveis com loadfiles encontrados na prática.

A transferência das configurações para a rede não estava inclusa inicialmente no projeto. Após as análises dos FOPs chegamos a conclusão que as interações com a base de dados não evidenciaria os defeitos que esperávamos.

5.2. Documentação do teste proposto

A estrutura do teste proposto segue principalmente a descrita por [Sommerville, 92] com algumas alterações para adaptar-se aos moldes dos planos de teste da AT&T Tridom.

A estrutura de um plano de testes segundo Sommerville é dividido em sete partes:

1. **O processo de teste:** Descrição das fases principais do processo do teste. O teste final ficou dividido em X fases: a) Instalação do CNCS; b) Configuração de uma rede com 10 VSATs na base de dados; c) Transferir as configurações para a rede; d) Transferir dados pela rede nos protocolos envolvidos.
2. **Requerimentos e seguimento.** Os parâmetros que determinarão os critérios de avaliação do processo de teste. Para o teste desejado não havia a necessidade de se ter

uma especificação muito rígida neste item. Para as interações com a base de dados, as alterações seriam aprovadas se fosse possível recuperá-la após o término da seção de configuração. Para as interações com os nós de rede, as alterações seriam aprovadas se fosse possível passar dados pelos mesmos nos protocolos para os quais foram configurados.

3. **Itens testados.** Escopo. As partes ou funções que serão testadas no software. Aqui entraram as configurações dos nós de rede que englobassem os protocolos mais usados. As configurações padrão estão em anexo.
4. **Cronograma de teste.** Uma noção geral de quanto tempo e recursos necessários para o teste. Conforme citado anteriormente, a meta seria ter um teste possível de ser executado em 10 dias por uma pessoa com treinamento básico da operação do CNCS.
5. **Procedimentos de registro do teste.** Esta seção descreve como os resultados do teste serão registrados. Neste trabalho tínhamos pontos de avaliação ao final da configuração de cada NPA. Ao final dos passos de adição, modificação ou exclusão de todos os recursos sob um NPA, as alterações eram salvas e a sessão concluída e reiniciada (procedimentos do CNCS).
6. **Requerimentos de hardware e software.** Descrição da configuração necessária para a realização do teste. Para a execução dos testes preliminares só foi preciso uma estação Unix para acessar o CNCS, já que não houve interações com a rede. No teste definitivo a especificação aumentou para uma rede de 10 VSATs completa.
7. **Restrições.** Elementos a serem observados a fim de não afetar os resultados do teste. A restrição mais séria foi que os endereços dos nós a serem testados estivessem sem nenhuma configuração ou nunca tivessem sido utilizados. Desta maneira poderíamos garantir que não se estava lidando com registros danificados previamente na base de dados. Tivemos esse problema durante os testes preliminares. Devido a um mal funcionamento de hardware antes dos testes começar, alguns registros foram danificados. Na base de dados do CNCS os registros são identificados pelo endereço do nó, que é único na partição. Ocorre que utilizamos um destes endereços já utilizados e danificados. Ao tentarmos alterar parâmetros do VSAT em questão, vários problemas surgiram e algum tempo foi gasto até que descobrissemos que não se tratavam de problemas do CNCS e sim dos registros da base de dados que estavam sendo utilizados.

Não nos foi permitido incluir o plano de testes final no presente relatório por ser parte da documentação proprietária da Tridom.

5.3. Teste preliminar e teste do CNCS 10.0

O teste desenvolvido foi aplicado na versão 9.4.1, como teste preliminar. A versão 9.4.1 já era considerada estável e havia sido lançada no mercado. Nesta versão foram detectados 6 erros e uma maneira de travar o CNCS.

Dois 6 erros encontrados, apenas dois foram dignos de se abrir FOPs. O primeiro tratava-se de uma tela de configuração de parâmetros do protocolo de acesso ao meio físico, TDMA, na

parte de definição dos horários relativos a cada configuração. Sempre que adicionávamos uma configuração, não era possível salvá-la. Outro foi um problema, mais sério, era relacionado a base de dados, que não alterava a configuração de alguns campos de uma link station de voz (VDLC).

O número de bugs “expressivos” identificados não foi tão grande nessa etapa preliminar, mas deve-se considerar que o software da versão testada já estava no mercado e vários defeitos já haviam sido encontrados e corrigidos. Após a aplicação do teste na versão 9.4.1 revisou-se o roteiro de testes e algumas modificações foram feitas:

1. O layout das folhas de teste, para podermos agilizar a anotação de resultados do teste
2. No passo de modificação passamos modificar as configurações do default para as sugeridas pelos developers

Em seguida montamos a bancada em uma partição isolada do laboratório do SVE e iniciamos o teste da versão 10.0 com a instalação da mesma no Stratus dedicado aos testes. A versão 10.0 trouxe diversas modificações de interface e na seção de alarmes que tiveram o formato alterado para funcionar após o ano 2000.

Ao total foram encontrados 5 erros que geraram FOPs e mais 3 erros menores referentes à padronização da interface ou da navegação pelos menus que foram desprezados em consenso com os orientadores do projeto.

O número de erros encontrados ficou aquém da meta de detecção de 20% dos número típico de erros encontrados em 6 meses de funcionamento de uma versão, que gira em torno de 30. O resultado atingido com o teste em questão foi um pouco menos de 17% (5 erros de um total estimado de 30). Mesmo assim um número significativo de bugs foram corrigidos antes da versão 10.0 ser lançado ao mercado, e quase o dobro do que seria detectado pelos métodos antigos, que em geral ficava abaixo de 3.

O teste proposto virou padrão para novas versões do CNCS.

6. Conclusões e Perspectivas

Os testes preliminares evidenciaram a necessidade do teste a nível de rede, e não só na base de dados. Na realidade o teste começou a funcionar - ou os defeitos começaram a aparecer - a partir do momento que começamos a tentar transferir as configurações para a rede e tentar transmitir dados do VSAT para o HI e vice-versa.

A principal sugestão apresentada ao orientador da empresa foi a padronização da decodificação dos comandos ODB. Isto possibilitaria a automação dos testes do início ao fim, que por sua vez possibilitaria um teste exaustivo de regressão para cada nova versão do CNCS.

Conforme exposto anteriormente os resultados foram positivos de maneira geral.

Durante a elaboração do trabalho identificou-se algumas falhas na padronização da interface com o usuário, entre estas maneiras diferentes de navegar entre os campos de determinadas telas e a apresentação dos campos em geral.

A tendência geral para a próxima geração de software de gerenciamento da rede é diminuir drasticamente o número de protocolos suportados. O número excessivo de versões do sistema se deve em parte ao número grande de soluções customizadas para clientes específicos. Pretende-se para a próxima arquitetura de rede suportar protocolos de maior popularidade entre o público alvo das redes VSAT como o TCP/IP, e X.25.

A evolução do hardware da rede VSAT deve facilitar a implementação de diversas aplicações sobre o TCP/IP, como voz e vídeo a 2 Mbps. Esta tendência de diminuir a complexidade do software deve simplificar muito os testes e a manutenção do produto.

7. Bibliografia

1. [Sommerville, 1992]
Sommerville I. *Software Engineering*. New York: Addison-Wesley Publishing, 4th Edition
2. [Boehm, 1979]
Boehm B. W. *Software Engineering: R & D Trends and Defense Needs. In research Directions of Software Engineering* (Wegner P., ed.). Cambridge MA: MIT Press.
3. [Myers, 1979]
Myers G. J. *The Art of Software Testing*. New York: John Wiley & Sons.
4. [Gillies, 1997]
Gillies A. C. *Software Quality: Theory and Management*. UK: T.J. International Ltd.
5. [Dyson, 95]
DYSON, P. *Dicionário de Redes*. Rio de Janeiro, Campus, 1995.
6. [Stemmer, 96] (usado como referência geral no início do projeto)
STEMMER, M. *Apostila do Curso de Redes de Comunicação de Dados*. Florianópolis, DAS-UFSC, 1996.
7. [Tridom 01]
AT&T Tridom. *Clearlink Network Architecture Overview*. Atlanta: AT&T Tridom, 1997
8. [Tridom 02]
AT&T Tridom. *Clearlink Networks Data Flow*. Atlanta: AT&T Tridom, 1997
9. [Tridom 03]
AT&T Tridom . *CNCS User's Guide*. Atlanta: AT&T Tridom, 1997
10. [Tridom 04]
AT&T Tridom. *CNCS Administrator's Guide*. Atlanta: AT&T Tridom, 1997
11. [Tridom 05]
AT&T Tridom. *Aegis User's Guide*. Atlanta: AT&T Tridom, 1997
12. [Tridom 06]
AT&T Tridom. *Clearlink X.25 Interface Technical Reference*. Atlanta: AT&T Tridom, 1997
13. [Tridom 07]
AT&T Tridom. *Clearlink Modem-PAD Technical Reference*. Atlanta: AT&T Tridom, 1997
14. [Tridom 08]
AT&T Tridom. *Loadfile name convention*. Atlanta: AT&T Tridom, 1997.
15. [Stratus, 90]

Stratus Inc. *Stratus - VOX Operarional System*. Los Angeles: Stratus Inc, 1990.

Referências na Internet:

<http://www.tridom.com>

<http://www.americom.com>

<http://www.spacenet.ge.com>

8. Anexos

8.1. Anexo 1: Configurações utilizadas no plano de testes final

Host Interface (HI)

```

Node #f-f7ff          is unchanged.
-NPA/1                is unchanged.
--TASK                MAILMAN is unchanged.
--TASK                IP       is unchanged.
--TASK                TP4      is unchanged.
--TASK                INET     is unchanged.
--TASK                ETHERNET is unchanged.
--TASK                VCS      is unchanged.
-NPA/2                is unchanged.
--TASK                MAILMAN is unchanged.
--TASK                MUX      is unchanged.
---PORT/0            MAP II MUX is unchanged.
----LINK/0           MUX      is unchanged.
-NPA/4                is unchanged.
--TASK                MAILMAN is unchanged.
--TASK                IP       is unchanged.
--TASK                LAPB     is unchanged.
---PORT/0            LAPB     is unchanged.
----LINK/0           LAPB     is unchanged.
---PORT/1            LAPB     is unchanged.
----LINK/0           LAPB     is unchanged.
---PORT/2            LAPB     is unchanged.
----LINK/0           LAPB     is unchanged.
---PORT/3            LAPB     is unchanged.
----LINK/0           LAPB     is unchanged.
---PORT/4            LAPB     is unchanged.
----LINK/0           LAPB     is unchanged.
---PORT/5            LAPB     is unchanged.
----LINK/0           LAPB     is unchanged.
---PORT/6            LAPB     is unchanged.
----LINK/0           LAPB     is unchanged.
---PORT/7            LAPB     is unchanged.
----LINK/0           LAPB     is unchanged.
-NPA/5                is unchanged.
--TASK                MAILMAN is unchanged.
--TASK                IP       is unchanged.
--TASK                BLAPB    is unchanged.
---PORT/0            BLAPB    is unchanged.
----LINK/0           BLAPB    is unchanged.
---PORT/1            BLAPB    is unchanged.
----LINK/0           BLAPB    is unchanged.
---PORT/2            BLAPB    is unchanged.
----LINK/0           BLAPB    is unchanged.
---PORT/3            BLAPB    is unchanged.
----LINK/0           BLAPB    is unchanged.
---PORT/4            BLAPB    is unchanged.
----LINK/0           BLAPB    is unchanged.
---PORT/5            BLAPB    is unchanged.
----LINK/0           BLAPB    is unchanged.
---PORT/6            BLAPB    is unchanged.
----LINK/0           BLAPB    is unchanged.
---PORT/7            BLAPB    is unchanged.
----LINK/0           BLAPB    is unchanged.
-NPA/18               is unchanged.
--TASK                MAILMAN is unchanged.
--TASK                IP       is unchanged.
--TASK                TLCC     is unchanged.
---PORT/3            MAP II CP is unchanged.
----LINK/0           TLCC     is unchanged.
---PORT/0            MAP II CP is unchanged.
----LINK/0           TLCC     is unchanged.
---PORT/1            MAP II CP is unchanged.
----LINK/0           TLCC     is unchanged.
---PORT/2            MAP II CP is unchanged.
----LINK/0           TLCC     is unchanged.
-NPA/19               is unchanged.
--TASK                MAILMAN is unchanged.
--TASK                IP       is unchanged.
--TASK                TLCC     is unchanged.
---PORT/0            MAP II CP is unchanged.

```

```

----LINK/0 TLCC is unchanged.
---PORT/2 MAP II CP is unchanged.
----LINK/0 TLCC is unchanged.
---PORT/1 MAP II CP is unchanged.
----LINK/0 TLCC is unchanged.
---PORT/3 MAP II CP is unchanged.
----LINK/0 TLCC is unchanged.
---PORT/4 MAP II CP is unchanged.
----LINK/0 TLCC is unchanged.

```

Switchig System (SS)

```

Node #F-F6FF is unchanged.
-NPA/1 is unchanged.
--TASK MAILMAN is unchanged.
--TASK IP is unchanged.
--TASK TP4 is unchanged.
--TASK X.25 L3 is unchanged.
--TASK GDS is unchanged.
--TASK INET is unchanged.
--TASK ETHERNET is unchanged.
--TASK VCS is unchanged.
---PORT/0 ETHERNET is unchanged.
-NPA/4 is unchanged.
--TASK MAILMAN is unchanged.
--TASK IP is unchanged.
--TASK TP4 is unchanged.
--TASK LAPB is unchanged.
---PORT/0 LAPB is unchanged.
----LINK/0 LAPB is unchanged.
-NPA/18 is unchanged.
--TASK MAILMAN is unchanged.
--TASK IP is unchanged.
--TASK TP4 is unchanged.
--TASK X.25 L3 is unchanged.
--TASK VDLC is unchanged.
---PORT/0 VDLC is unchanged.
----LINK/0 VDLC is unchanged.
-NPA/19 is unchanged.
--TASK MAILMAN is unchanged.
--TASK IP is unchanged.
--TASK TP4 is unchanged.
--TASK ConSrvr is unchanged.
--TASK HYDRA is unchanged.
---PORT/0 HYDRA is unchanged.
----LINK/0 HYDRA is unchanged.
-----LS/0 CLPAD is unchanged.
---PORT/1 HYDRA is unchanged.
----LINK/0 HYDRA is unchanged.
-----LS/0 MPAD is unchanged.
-NPA/20 is unchanged.
--TASK MAILMAN is unchanged.
--TASK IP is unchanged.
--TASK TP4 is unchanged.
--TASK ConSrvr is unchanged.
--TASK X.25 L3 is unchanged.
---PORT/0 X.25 L3 is unchanged.
----LINK/0 X.25 L3 is unchanged.
---PORT/1 X.25 L3 is unchanged.
----LINK/0 X.25 L3 is unchanged.
-NPA/22 is unchanged.
--TASK MAILMAN is unchanged.
--TASK IP is unchanged.
--TASK TP4 is unchanged.
--TASK CVCC is unchanged.
---PORT/0 SDLC is unchanged.
----LINK/0 SDLC is unchanged.
-----LS/0 SDLC is unchanged.
-----LS/1 SDLC is unchanged.
-----LS/2 SDLC is unchanged.
-----LS/3 SDLC is unchanged.
-----LS/4 SDLC is unchanged.
-----LS/5 SDLC is unchanged.
-----LS/6 SDLC is unchanged.
-----LS/7 SDLC is unchanged.
-----LS/8 SDLC is unchanged.
-----LS/9 SDLC is unchanged.
---PORT/2 SDLC is unchanged.
----LINK/0 SDLC is unchanged.

```

```

-----LS/0   SDLC      is unchanged.
-----LS/1   SDLC      is unchanged.
-----LS/2   SDLC      is unchanged.
-----LS/3   SDLC      is unchanged.
-----LS/4   SDLC      is unchanged.
-NPA/26      is unchanged.
--TASK       MAILMAN   is unchanged.
--TASK       IP         is unchanged.
--TASK       TP4        is unchanged.
--TASK       3780BSC    is unchanged.
---PORT/0    3780 BSC    is unchanged.
----LINK/0   3780 BSC    is unchanged.
----LS/0     3780 BSC    is unchanged.

```

VSAT 1 - Boot File SXHv+100 [Tridom 08]

Node f-ff7f

```

-NPA 1       VSAT
--TASK       MAILMAN
--TASK       IP
--TASK       TP4
--TASK       TLCS
---PORT 0    MAP II CS
----LINK 0   TLCS
---PORT 1    SDLC
----LINK 0   SDLC
----LS/0     SDLC
---PORT 2    CSO Plus
---PORT 3    3 X.25 L3
----LINK 0   X.25 L3
---PORT 5    HYDRA
----LINK 0   HYDRA
----LS 0     CLPAD

```

VSAT 2 - f-ff9f Boot File Dsv+100

Node #f-ff9f

```

-NPA/1
--TASK       MAILMAN
--TASK       IP
--TASK       TP4
--TASK       CVCC
--TASK       LABP
--TASK       GDS
--TASK       TLCS
--TASK       TNET
--TASK       TRMR
---PORT 0    MAP II CS
----LINK 0   TLCC
---PORT 1    SDLC
----LINK 0   SDLC
----LS/0     SDLC
----LS/1     SDLC
---PORT 2    CSO PLUS
----LINK 0   LAPB
---PORT 4    TDLS
----LINK 0   TDLS
---PORT 5    TNET
----LINK 0   SDLC

```

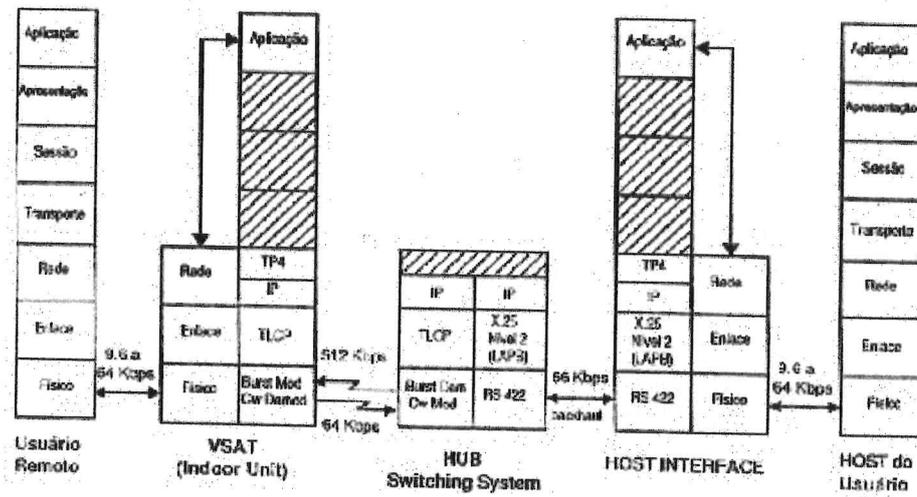
VSAT 3 - f-ff9a Boot File IRvic30e

```

Node #f-ff9a
-NPA/1       is unchanged.
--TASK       MAILMAN   is unchanged.
--TASK       IP         is unchanged.
--TASK       TP4        is unchanged.
--TASK       ConSrvr    is unchanged.
--TASK       X.25 L3     is unchanged.
--TASK       X.25ATP    is unchanged.
--TASK       TLCS       is unchanged.
---PORT/0    MAP II CS
----LINK/0   TLCS
---PORT/1    X.25 ATP
----LINK/0   X.25 ATP

```

8.2. Camadas OSI na rede Clearlink



8.3. Anexo 3. Visão geral dos processos e suas localização na arquitetura Clearlink.

[Tridom 02]

