UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

ESCOLA POLITÉCNICA

DEPARTAMENTO DE ELETRÔNICA E DE COMPUTAÇÃO

PROTOCOLO DE COMUNICAÇÃO BASEADO NA NORMA IEC 870-5-101

Autor: _____
Bruno Garrofé Sampaio

Orientador : _____
Prof. Aloysio de Castro P. Pedroza

Examinador : _____
Prof. Edilberto Strauss

Examinador : _____
Prof. José Paulo Brafman

DEL

Outubro de 2003

Dedico este trabalho a Elizabeth, Valmir, Mariana, Henrique, Ilza, Paulo, Rosana, Henrique, Liana, Ângela, José Luiz, André e Luísa.

AGRADECIMENTOS

A Deus, que sempre esteve presente na minha vida.

À Mariana, por oferecer todo o incentivo, ajuda e paciência.

Aos meus pais e familiares, por toda a educação e dedicação investida na minha formação.

Aos amigos, que sempre estiveram por perto nos melhores e piores momentos.

Ao professor Aloysio pela orientação transmitida.

Ao Sérgio Rohenkohl e toda a equipe de Furnas, por terem permitido, autorizado e incentivado a realização deste trabalho.

RESUMO

Este projeto analisa o desenvolvimento de um protocolo de comunicação baseado na norma IEC 870-5-101. A partir das definições apresentadas nas normas às quais o protocolo faz referência, foi desenvolvido um aplicativo visando implementar as funcionalidades de um centro de supervisão e controle no desempenho de tarefas de telemetria. Este protocolo visa monitorar e gerenciar remotamente, subestações e usinas que compõem um sistema de transmissão e distribuição de energia elétrica.

SUMÁRIO

AGRADECIMENTOS.....iv
RESUMO.....v
CAPÍTULO I – INTRODUÇÃO.....8
CAPÍTULO II – EMBASAMENTO TEÓRICO.....10
II.1 – Motivação.....10
II.2 – Protocolos de Comunicação.....14
II.2.1 – Camada física.....17
II.2.2 – Camada de link.....17
II.2.3 – Camada de aplicação.....18
II.2.4 – Modelagem do protocolo.....18
II.3 – As Normas.....19
II.3.1 - IEC 870-5-1.....22
II.3.1.1 - Requisitos para a transmissão de dados.....24
II.3.1.1.1 - Alta integridade e consistência dos dados.....24
II.3.1.1.2 - Curto tempo de transmissão de comandos.....24
II.3.1.2 - Classes de integridade.....25
II.3.1.3 - Classes de serviço na camada de link.....25
II.3.1.4 - Padrões de formato de frames.....27
II.3.2 - IEC 870-5-2.....28
II.3.2.1 – Formatos de frame.....29
II.3.2.1.1 – Formato FT 1.1.....29
II.3.2.1.2 - Formato FT 1.2.....30
II.3.2.1.3 - Formato FT 2.....31
II.3.2.1.4 - Formato FT 3.....33
II.3.2.2 – Tipos de Transmissão de Mensagens.....34
II.3.2.3 – Byte de Controle.....34
II.3.2.4 – Classes de mensagens de dados.....37
II.3.2.5 – Byte de Endereço.....38
II.3.3 – IEC 870-5-3.....39
II.3.3.1 – Estrutura de Dados de Aplicação.....40
II.3.3.1.1 – Unidade de dados de serviço da aplicação (ASDU).....41
II.3.3.1.2 – Identificador de unidade de dados.....42

II.3.3.1.3 – Objetos de informação....43
II.3.3.1.4 - Identificação dos objetos de informação....45
II.3.3.1.5 – Seqüências de elementos de informação....46
II.3.4. IEC 870-5-4....47
II.3.5. IEC 870-5-5....48
II.3.5.1 – Inicialização da estação....51
II.3.5.1.1 – Inicialização da estação de controle....52
II.3.5.1.2 – Inicialização da estação remota....52
II.3.5.2 – Inicialização local da estação de controle em sistemas de transmissão não-balanceados....54
II.3.5.3 – Inicialização local da estação remota em sistemas de transmissão não-balanceados....56
II.3.5.4 – Inicialização remota da estação remota em sistemas de transmissão não-balanceados....58
II.3.5.5 - Aquisição de dados por "polling"....60
II.3.5.6 – "General Interrogation"....62
II.3.5.7 – Sincronismo do Clock....64
II.3.5.8 – Transmissão de comandos....66
II.3.6. IEC 870-5-101....69
II.3.6.1 – Camada de link....69
II.3.6.2 – Camada de aplicação....71
II.3.6.2.1 – Identificador de tipo....73
II.3.6.2.2 – Qualificador variável de estrutura....74
II.3.6.2.3 – Causa da transmissão....75
II.3.6.2.4 – Endereço comum da ASDU....76
II.3.6.2.5 – Endereço do objeto de informação....77
II.3.6.3 – Unidades de Dados de Serviço de Aplicação (ASDUs)....78
II.3.6.3.1 – Informação de estado sem tag de tempo (M_SP_NA_1), SQ = 0....79
II.3.6.3.2 – Informação de estado sem tag de tempo (M_SP_NA_1), SQ = 1....80
II.3.6.3.3 – Informação de estado com tag de tempo (M_SP_TA_1)....81
II.3.6.3.4 – Informação de estado duplo sem tag de tempo (M_DP_NA_1), SQ = 0...82
II.3.6.3.5 – Informação de estado duplo sem tag de tempo (M_DP_NA_1), SQ = 1...83
II.3.6.3.6 – Informação de estado duplo com tag de tempo (M_DP_TA_1)....84
II.3.6.3.7 – Informação de medida normalizada (M_ME_NA_1), SQ=0....85

II.3.6.3.8 – Informação de medida normalizada (M_ME_NA_1), SQ=1.....86
II.3.6.3.9 – Informação de medida normalizada com tag de tempo (M_ME_TA_1)...87
II.3.6.3.10 – Comando simples (C_SC_NA_1).....88
II.3.6.3.11 – Comando duplo (C_DC_NA_1).....89
II.3.6.3.12 – Comando de "general interrogation" (C_IC_NA_1).....90
II.3.6.3.13 – Comando de sincronismo do clock (C_CS_NA_1).....91
II.3.6.3.14 – Comando de teste (C_TS_NA_1).....92
CAPÍTULO III – METODOLOGIA DE IMPLEMENTAÇÃO.....93
III.1 – Considerações gerais.....94
III.2 – A função principal "main".....95
III.3 – A função "Espera_e_trata_eventos".....96
III.3.1 – A função "Inicializa_Var".....96
III.3.2 – A função "Monta_msg_IEC_tam_fixo".....98
III.3.3 – O byte de verificação ou checksum.....100
III.3.4 – A função de transmissão "Transmite_msg_IEC".....101
III.3.5 – O estado de espera pela mensagem de resposta.....102
III.3.6 – Ocorrência de timeout.....102
III.3.7 – As condições de recebimento de mensagens.....104
III.3.8 – A função "Verifica_msg_IEC".....104
III.3.9 – A função "Ocorreu_erro".....105
III.3.10 – A função "Trata_byte_controle".....106
III.3.11 – As mensagens de tamanho variável.....108
III.3.12 – A camada de aplicação.....108
CAPÍTULO IV – TESTES.....111
IV.1 – Ativação da remota.....112
IV.2 – Remota fora de serviço.....113
IV.3 – Aquisição de dados.....115
IV.4 – Envio de "general interrogation".....116
IV.5 – Envio de sincronismo de clock.....119
IV.6 – Envio de comandos.....120
CAPÍTULO V – CONCLUSÃO.....122
REFERÊNCIAS BIBLIOGRÁFICAS.....124

CAPÍTULO I – INTRODUÇÃO

Os protocolos são peças fundamentais no processo de comunicação, pois são responsáveis por estabelecer regras e normas para que a comunicação entre as partes envolvidas seja eficiente e correta.

Os protocolos estão presentes mesmo quando não se aborda a área de computação. Para que duas pessoas possam se comunicar entre si é necessário que ambas obedeçam a regras. Para o processo da fala, por exemplo, essas regras são as gramaticais e as semânticas, que por sua vez possuem características diferentes em cada língua. Assim como cada língua possui o seu conjunto de normas, da mesma maneira ocorre com os protocolos. Para que dois computadores ou equipamentos possam "conversar" entre si é preciso que estes estejam "falando a mesma língua".

A importância dos protocolos nos dias de hoje se dá devido à enorme quantidade de informações diferentes, que precisam ser intercambiadas entre as diversas pessoas que compõem uma sociedade. Cada vez mais os elementos desta sociedade se aproximam através da tecnologia.

O protocolo aqui apresentado e desenvolvido irá utilizar normas e regras definidas segundo a IEC (International Eletrotechnical Commission) com o objetivo de criar um aplicativo capaz de estabelecer a comunicação entre dois dispositivos, de forma a implementar um processo de supervisão e controle. Busca-se controlar e supervisionar remotamente subestações e equipamentos que compõem um sistema de transmissão de energia elétrica.

Inicialmente, será apresentado o cenário que tornou necessário o desenvolvimento do protocolo em questão, ou seja, serão mostradas as condições que levaram a decisão de desenvolver este protocolo especificamente. Posteriormente, apresentar-se-ão seus conceitos básicos, salientando as características mais comuns dos protocolos, para depois descrever mais especificamente, as normas que o compõem.

No segundo capítulo apresentar-se-á a metodologia de implementação, ilustrando como o programa foi desenvolvido e como este está estruturado. Finalmente, serão ilustrados alguns testes realizados, de modo a verificar o comportamento do programa em diversas situações diferentes.

CAPÍTULO II – EMBASAMENTO TEÓRICO

Neste capítulo apresentar-se-á o cenário no qual o protocolo está inserido, ou seja, serão explicadas as condições que tornaram necessário o desenvolvimento deste protocolo. Posteriormente, apresentar-se-ão os aspectos fundamentais que caracterizam um protocolo de comunicação, mostrando como este pode ser dividido em módulos independentes, com características próprias e distintas entre si.

Será também proposto um modelo comportamental do protocolo, para que em seguida se possa dar início ao detalhamento das normas que regem o programa desenvolvido. Por fim, serão apresentadas as normas que compõem o protocolo IEC 870-5-101.

II.1 – Motivação

A diretoria de sistemas de supervisão e controle de Furnas, também conhecida como DSSC, é responsável pelo monitoramento dos elementos que compõem a rede de transmissão de energia elétrica de Furnas, que por sua vez possui diversas subestações e usinas geradoras de energia espalhadas por vários estados do Brasil. Cada subestação ou usina é composta por muitos elementos, como linhas de transmissão de alta tensão, chaves secionadoras, barramentos, transformadores, reatores e capacitores. Todos os elementos precisam estar funcionando corretamente para que a rede, como um todo, opere segundo a demanda energética das empresas distribuidoras de energia e, conseqüentemente dos consumidores.

Com o objetivo de garantir o correto funcionamento de todos os elementos que compõem esta vasta rede de distribuição elétrica de Furnas, foi desenvolvido um centro de supervisão e

controle responsável por monitorar e gerenciar o funcionamento de usinas e subestações. Este centro está fisicamente interligado com as partes integrantes do sistema elétrico, coletando dados operacionais, como tensão e corrente das linhas de transmissão, estados das chaves secionadoras, potência gerada, assim como outras informações consideradas importantes.

Além de obter informações, o centro de supervisão e controle também pode enviar comandos, por exemplo, com o intuito de mudar a tensão ou corrente em uma linha de transmissão, ou também, para realizar uma manobra de manutenção em um equipamento. Todos esses procedimentos podem ser realizados à distância, graças à presença de máquinas especializadas no desempenho destas tarefas, assim como ao advento das telecomunicações, que permitiu que dados fossem transportados através de longas distâncias.

Para que as informações possam ser enviadas para o centro de supervisão e controle, existem equipamentos especializados, presentes nas subestações e usinas, responsáveis por coletar todas as informações importantes, armazenando-as para que possam ser transmitidas quando solicitado pelo centro de controle.

Os equipamentos especializados na transmissão de dados podem ser fabricados por diversas empresas diferentes, que poderiam adotar cada uma um padrão diferente de transmissão. Como isso seria prejudicial tanto para o fornecedor quanto para o consumidor, algumas organizações internacionais desenvolveram regras e normas que deveriam ser seguidas, de forma a padronizar o formato dos dados transmitidos, assim como o modo de transmissão.

Desta forma, empresas distintas poderiam utilizar seus próprios processos de produção, porém, a etapa final correspondente à transmissão e formatação dos dados transmitidos seria

padronizada seguindo as regras definidas em uma determinada norma, e assim empresas diferentes teriam a garantia de que se ambas estivessem utilizando a mesma norma, conseguiriam se comunicar. Essas regras utilizadas para transmitir informações são conhecidas como protocolo.

Com o passar dos anos e com a evolução tecnológica, alguns protocolos vão sendo substituídos por outros devido à existência de vantagens que seus antecessores não possuíam. Assim, muitas vezes um centro de supervisão e controle tem que lidar com mais de um protocolo de comunicação.

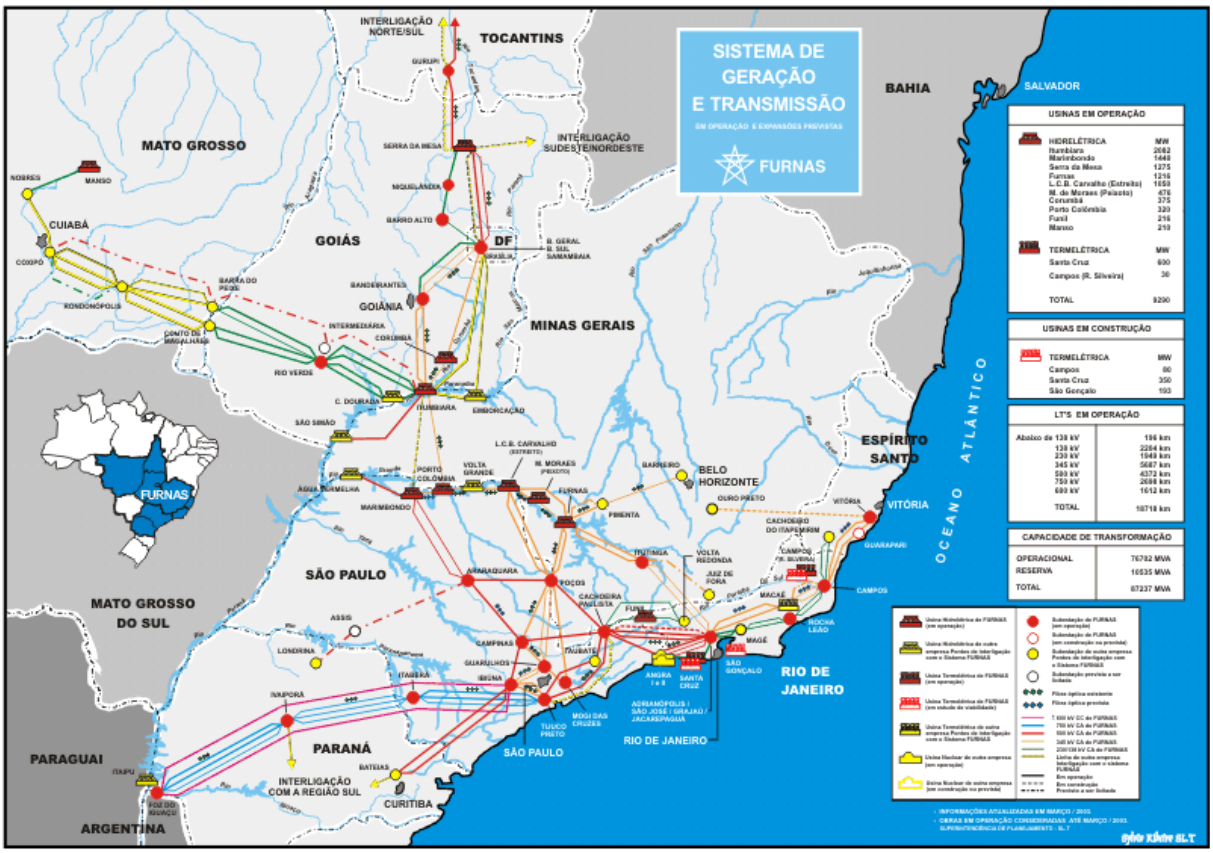
Este é o caso de Furnas, que possui programas os quais implementam protocolos mais antigos para permitir a compatibilidade com subestações ou usinas que utilizam métodos de transmissão mais antigos. Além disso, também possui programas executando protocolos modernos utilizados na comunicação com novos equipamentos, presentes por exemplo em usinas que ainda serão construídas ou subestações reformadas.

A proposta para o desenvolvimento do protocolo IEC 870-5-101 surgiu devido à crescente utilização deste pelos fabricantes de equipamentos de transmissão nas usinas e subestações. Como estagiário, fui membro integrante da equipe de engenheiros de Furnas, responsável pelo desenvolvimento de um software visando à implementação deste protocolo. Após sua conclusão, foi devidamente testado, através de aplicativos comerciais especializados, além de terem sido feitos testes reais em subestações. Tendo sido aprovado em todos os testes este se encontra, no presente momento, em operação, monitorando e supervisionando subestações e usinas que compõem o sistema elétrico de Furnas.

Vale ressaltar, que o programa desenvolvido em Furnas foi escrito na linguagem de programação C, em computadores da Digital/Compaq, que rodam um sistema operacional proprietário conhecido como OpenVMS. Este sistema operacional possui algumas particularidades e funções específicas, que foram utilizadas no desenvolvimento do programa.

O programa apresentado neste trabalho foi adaptado para o sistema operacional Linux, particularmente desenvolvido na versão 7.3, para que pudesse ser testado e utilizado mais facilmente em computadores pessoais. Desta forma, o programa original sofreu algumas alterações visando torná-lo compatível com este novo ambiente.

Figura II.1 – Sistema de Geração e Transmissão de Furnas



Fonte: www.furnas.com.br

II.2 – Protocolos de Comunicação

Um protocolo de comunicação nada mais é do que um "**conjunto de convenções que rege o tratamento e, especialmente, a formatação de dados num sistema de comunicação**".

Podemos definir um protocolo de comunicação de dados como um conjunto de regras que controla a comunicação para que a mesma seja eficiente e sem erros. Um dos objetivos principais do protocolo é detectar e evitar a perda de dados ao longo da transmissão, solicitando a retransmissão dos mesmos, caso isto ocorra. O protocolo nada mais é que um software ou programa de computador, que recebe ou envia dados a serem transmitidos, agregando no início e no fim das mensagens transmitidas os caracteres de controle, confirmação de recebimento, controle de seqüência das mensagens ou blocos de dados transmitidos, cálculo do algoritmo de detecção de erros e outros controles necessários a uma boa transmissão

Os protocolos de comunicação de dados são fundamentais e imprescindíveis para que ocorra uma transferência de dados com segurança. Sem o uso dos mesmos perdemos a integridade dos dados ao longo da transmissão, inviabilizando a comunicação entre os programas, computadores e equipamentos. O protocolo é um programa carregado no computador e agregado às interfaces de comunicação do mesmo, com o objetivo básico de garantir que um dado qualquer possa ser recebido da mesma forma que foi transmitido. A integridade dos dados é mantida, independente do meio utilizado na transmissão já que o controle é feito nas pontas.

De forma a melhor caracterizar os diferentes tipos de protocolos quanto às funções que desempenham, foi criado o modelo de referencia OSI (Open Systems Interconnection). Este modelo descreve como uma informação proveniente de uma aplicação em um computador trafega através da rede até a outra aplicação em outro computador. O modelo OSI é um modelo conceitual composto de 7 camadas, cada uma responsável por especificar uma função particular da rede. Este modelo foi desenvolvido pela ISO (International Organization for Standardization) em 1984 e é atualmente considerado o modelo de arquitetura mais importante para a comunicação entre computadores. Cada uma das camadas que compõem o modelo são independentes entre si, permitindo que soluções desenvolvidas para uma camada sejam atualizadas sem afetar as outras camadas.

Figura II.2 – Modelo OSI

	Camada
APLICAÇÃO	7
APRESENTAÇÃO	6
SEÇÃO	5
TRANSPORTE	4
REDE	3
LINK	2
FÍSICA	1

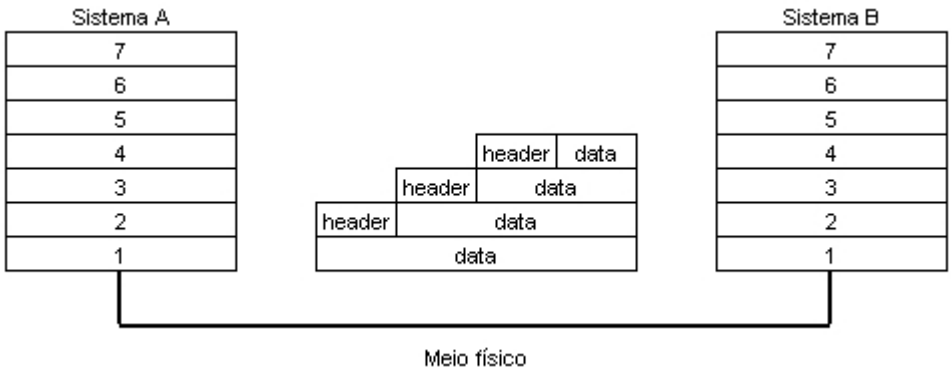
As camadas mais elevadas do modelo tratam de questões de aplicação e normalmente somente são implementados por software. A camada de aplicação é a mais próxima do usuário. As camadas mais baixas lidam com questões relativas ao transporte dos dados. A camada física e de link são implementadas em software e em hardware. A mais baixa, a física, é a que está mais próxima do meio de transmissão e é responsável por transferir a informação para este.

O modelo OSI possibilita uma visualização conceitual do processo de comunicação entre os computadores, mas o modelo por si próprio não é um método de comunicação. Um protocolo pode implementar as funções de uma ou mais camadas do modelo OSI. Este é o caso do protocolo baseado na norma IEC 870-5-101 que iremos apresentar nas seções seguintes, pois este é definido nas camadas de aplicação, de link e física.

A informação a ser transferida de uma aplicação em computador para outra aplicação em outro computador precisa passar por todas as camadas definidas no modelo OSI. As camadas utilizam informações de controle para se comunicarem com as camadas de mesmo nível no computador remoto. Estas informações de controle consistem de pedidos específicos e instruções que são trocadas entre as camadas do modelo.

As informações de controle normalmente são caracterizadas por "headers" e "trailers". Os "headers" são posicionados antes da informação que foi passada da camada superior para a camada atual, enquanto que os "trailers" são posicionados após esta informação. Esta ação de adicionar informações de controle antes a após os dados enviados pelas camadas superiores é chamada de encapsular do dado.

Figura II.3 – Os dados de uma camada são encapsulados na camada inferior



A seguir, iremos descrever as características gerais das camadas que serão apresentadas posteriormente nas seções seguintes.

II.2.1 – Camada física

A camada física define as especificações elétricas, mecânicas e funcionais para a ativação, manutenção e desativação do link físico entre os sistemas de comunicação. Estas especificações definem características como: níveis de voltagem, temporização das mudanças de voltagem, taxas de transmissão, distancias máximas de transmissão e conectores físicos.

II.2.2 – Camada de link

A camada de link promove o trânsito íntegro de dados através de um link físico. Diferentes especificações para a camada de link definem diferentes características de protocolo e de rede, incluindo endereçamento físico, topologia da rede, notificação de erro, seqüênciamento dos frames e controle de fluxo de dados. O endereçamento físico define como os dispositivos são endereçados na camada de link. A topologia da rede consiste de especificações que normalmente definem como os dispositivos serão fisicamente conectados, como em um barramento ou em anel. Notificação de erro alerta as camadas superiores que ocorreu um erro de transmissão e o senqüênciamento de frames reordena os frames que são transmitidos fora da seqüência. Finalmente, o controle de fluxo controla a transmissão de dados para que o receptor não sofra overflow.

II.2.3 – Camada de aplicação

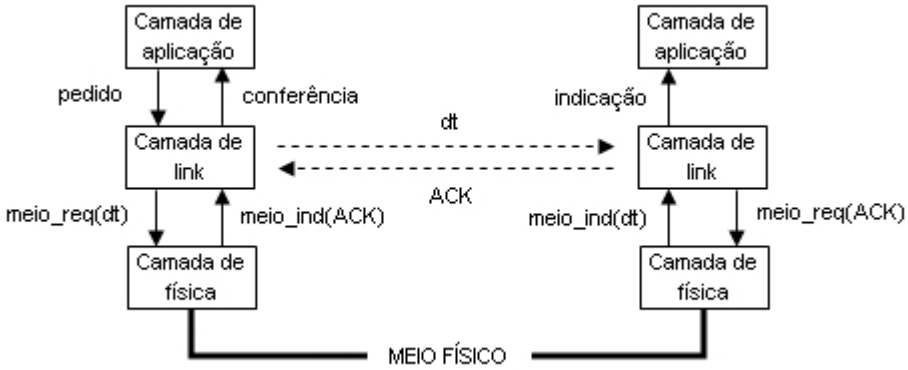
A camada de aplicação é a camada que está mais próxima do usuário o que significa que tanto esta camada quanto o usuário interagem diretamente com a aplicação. As funções típicas desta camada incluem a identificação de padrões de comunicação, determinação de disponibilidade de recursos e o sincronismo da comunicação.

II.2.4 – Modelagem do protocolo

O protocolo de comunicação que será desenvolvido irá se basear no conceito de máquinas de estado. Isto se dá devido à característica assíncrona da transmissão onde uma das pontas envia pedidos ou comandos para a outra ponta. Esta somente poderá enviar mensagens em resposta às solicitações da primeira. Este processo será explicado com mais detalhes posteriormente, porém já se pode propor um modelo em rede de Petri dos processos de comunicação de dados envolvidos.

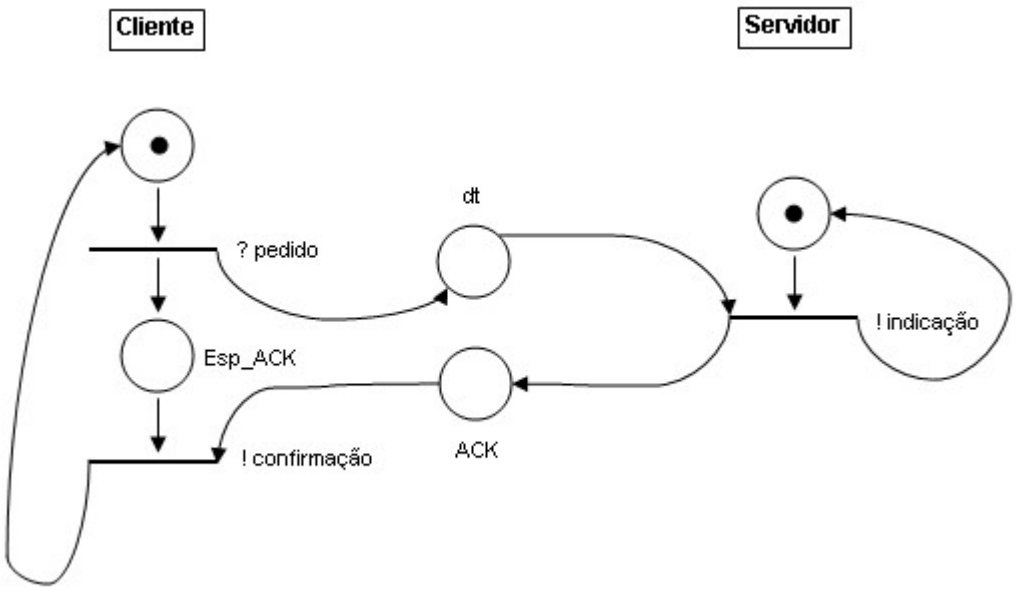
Como vimos anteriormente, o protocolo desenvolvido irá utilizar 3 camadas do modelo OSI: camada de aplicação, camada de link e camada física. Estas camadas irão estabelecer comunicação entre si segundo o modelo mostrado abaixo.

Figura II.4 – Modelo de troca de mensagens entre dois sistemas



Desta forma podemos estabelecer o diagrama de comunicação entre os processos do protocolo da seguinte forma.

Figura II.5 – Modelo em rede de Petri para comunicação entre dois sistemas



II.3 – As Normas

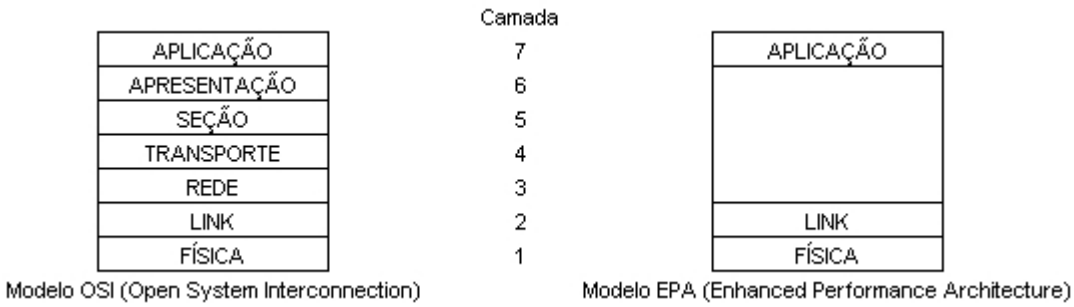
O protocolo baseado na norma IEC 870-5-101 baseia-se nas normas que serão descritas a seguir, e por isso é necessário que estas sejam explicadas para o correto entendimento do

funcionamento do mesmo. Algumas destas normas, por serem mais relevantes no desenvolvimento do protocolo, serão discutidas com mais detalhamento enquanto que outras serão descritas mais superficialmente. A seguir, iremos realizar um breve resumo sobre o contexto de cada uma das normas analisadas, para em seguida nos aprofundarmos nos assuntos mais importantes de cada uma delas.

A primeira norma usada como base para a implementação do protocolo é a IEC 870-5-1. Esta norma discorre sobre os formatos ou tipos de frames de transmissão, elucida requisitos e condições específicas para a transmissão de dados em sistemas de telemetria e mostra maneiras de atingir estas especificações.

Em termos do modelo de referencia OSI (Open System Interconnection), o qual divide o processo de comunicação em sete camadas, podemos situar esta norma nas duas camadas mais baixas a saber, a camada física e a camada de link, como pode ser visto na figura abaixo. O modelo EPA é uma especialização do OSI para sistemas que necessitam de um pequeno tempo de resposta, com banda de transmissão reduzida.

Figura II.6 – O modelo OSI e o modelo EPA



A segunda norma utilizada é a IEC 870-5-2. Nesta norma, são descritos os processos de transmissão da camada de link, assim como estruturas de transmissão de dados para a camada

de link. A norma citada anteriormente estava situada nas duas camadas mais baixas do modelo OSI. Esta, por sua vez, também pode ser localizada na segunda camada deste modelo, complementando as definições da outra norma. Nesta seção será descrita a disposição dos dados dentro dos frames de transmissão sob a regência de certas estruturas. São indicados também, tipos de mensagens que são trocadas entre as camadas de link de cada uma das duas pontas do link de transmissão.

Na terceira norma descrita, a IEC 870-5-3, serão definidas as estruturas genéricas para os dados de nível de aplicação nos frames de transmissão. Esta seção determina regras para a estruturação de dados de nível de aplicação. Estas regras são apresentadas como padrões genéricos que podem ser usados para dar suporte a uma grande variedade de aplicações presentes e futuras na área de telecomunicações. Esta norma pode ser alocada na sétima camada do modelo de referencia OSI.

A quarta norma apresentada é a IEC 870-5-4 que define regras sobre os tipos de elementos de dados utilizados nas trocas de mensagens. Dentre outras atribuições, esta regra procura padronizar os tipos de dados e busca definir tipos únicos utilizados para representar elementos digitais, como por exemplo o estado de uma chave digital, ou elementos analógicos, como a medida da tensão em uma linha de transmissão, entre outros elementos.

A quinta norma, a IEC 870-5-5 estabelece algumas funções básicas no nível de aplicação que serão utilizadas nos procedimentos de controle de sistemas de telemetria e controle à distância. São definidas também algumas regras para estabelecer uma adequada troca de mensagens no nível de aplicação entre o processo local e o processo remoto, de forma a garantir a aquisição de dados da remota, assim como o envio de comandos à mesma.

Por último, serão vistos os aspectos contidos na norma IEC 870-5-101 que define funções com o objetivo de garantir a interoperabilidade entre equipamentos de telemetria. Estas funções são utilizadas para realizar diversas tarefas, como a aquisição de dados, envio de comandos de controle, sincronização entre os dois equipamentos, comandos para teste de link entre outros.

A seguir, detalharemos cada uma destas normas ressaltando os pontos mais importantes considerados na elaboração do protocolo de comunicação, focando sempre na elucidação dos processos desempenhados pelo protocolo. Alguns detalhes presentes na norma poderão ser omitidos para fins de compatibilidade com a proposta de desenvolvimento ou para manter a clareza.

II.3.1 - IEC 870-5-1

Como já descrito anteriormente, nesta seção iremos discutir sobre os formatos para transmissão de dados através de frames.

O propósito maior das funções de comunicação nos processos de monitoração e controle está em atingir a consistência máxima, isto é, não deve haver discrepâncias entre os estados físicos das variáveis de processo e suas respectivas imagens no banco de dados do sistema de telemetria. Como exemplo, podemos citar um sistema de controle de uma usina hidroelétrica. A aquisição de dados realizada por um centro de controle que esteja monitorando a atividade da usina necessita ser extremamente confiável, ou seja, os valores de tensão ou corrente fornecidos pela usina precisam ser os mesmos valores visto por um operador do sistema, para que as medidas adequadas possam ser tomadas sem causar maiores danos.

Este objetivo de atingir confiabilidade total não pode ser atingido devido às leis da causalidade. Estas determinam que informações sobre variáveis de processo podem ser atrasadas, e também que valores reais possam ser distorcidos devido a ruídos ou distúrbios causados por falhas ou por fatores desconhecidos ou fora de controle. O que podemos esperar é que os processos de comunicação permitam a manutenção de um alto grau de consistência.

Em um ambiente não ideal, um alto nível de integridade e a eficiente transmissão da informação são fatores conflitantes. Por exemplo, um aumento na necessidade de elevada consistência da informação pode acarretar em uma redução na velocidade de transmissão uma vez que um número maior de bits precisa ser inserido na mensagem transmitida para garantir a detecção de erros de transmissão. Desta forma, é importante achar um compromisso entre estes dois fatores, baseado na análise dos requisitos dos sistemas e dos projetos.

O transporte de dados representa apenas um dos aspectos importantes na formação do sistema. De acordo com a norma, três classes de fatores podem ser descritas para garantir a correta definição do protocolo a ser utilizado e da forma mais adequada de fazê-lo. Essas classes são os requisitos funcionais, as características da rede e as condições da planta. Os requisitos funcionais podem ser divididos em: integridade dos dados, disponibilidade do sistema, tempo de resposta e acurácia. Outra classe importante é a característica da rede que engloba a configuração da rede, banda de transmissão e taxa de erro. Por último, é importante que seja levado em consideração as condições da planta e seus fatores como: situação geográfica, número e tipos de mensagens suportadas entre diversos outros fatores.

II.3.1.1 - Requisitos para a transmissão de dados

De forma a atingir os objetivos dos sistemas de controle e telemetria, os seguintes requisitos precisam ser preenchidos:

II.3.1.1.1 - Alta integridade e consistência dos dados

A correta transmissão de dados é necessária na presença de condições adversas tanto físicas como ambientais, como interferência eletromagnética, diferenças de potencial terrestre, envelhecimento de peças e componentes, assim como outras fontes de distúrbios e interferências no caminho da transmissão. Sob estas condições se faz importante proteger as mensagens contra os seguintes problemas:

- erros não detectados nos bit recebidos;
- erros não detectados nos frames, causado por erro de sincronismo;
- perdas não detectadas de informação;
- aquisição de informação não desejada (através de ruídos na linha); ou
- separação ou perturbação de informação coerente.

II.3.1.1.2 - Curto tempo de transmissão de comandos

Fornecimento de dados em um curto período de tempo através da utilização de eficientes frames de transmissão, assim como protocolos eficientes, particularmente para mensagens espontâneas geradas em meios de transmissão com banda limitada e com características incertas de ruído.

II.3.1.2 - Classes de integridade

Três diferentes classes de integridade foram definidas para caracterizar a transmissão de dados, a saber: I1, I2 e I3. O uso de cada uma delas depende da natureza do dado o do tipo de tratamento que se deseja dar a este. Normalmente, as transmissões feitas utilizando a classe I1 são destinadas a atividades pouco importantes, ou a processos cíclicos que demandam baixo nível de integridade. As transmissões feitas na classe I2 se destinam a transmissões com prioridade mais alta e são utilizadas em processos de telemetria. As mensagens pertencentes à classe I3 são classificadas como críticas ou de altíssima prioridade. Por exemplo, vamos supor que um link de transmissão apresente taxa de erro de bit igual a 10^{-4}. Uma mensagem transmitida utilizando-se a classe I1 apresentaria 1 erro não detectado a cada 1 dia. Por sua vez, uma mensagem transportada segundo a classe I2 apresentaria 1 erro a cada 26 anos e por fim na classe I3, 1 erro não detectado só ocorreria uma vez a cada 260.000 anos. Desta forma, podemos perceber que à medida que necessitamos transportar mensagens com segurança teremos que utilizar mensagens com classes de integridade apropriada. Da mesma forma, não podemos esquecer do compromisso entre integridade e tamanho da mensagem. À medida que aumentamos a integridade, aumentamos também o tamanho das mensagens.

II.3.1.3 - Classes de serviço na camada de link

Podemos definir três classes de serviço utilizadas pela camada de link:

- S1, SEND/NO REPLY
- S2, SEND/CONFIRM
- S3, REQUEST/RESPOND

A classe de serviço S1, SEND/NO REPLY é utilizada em sistemas cíclicos de atualização ou em sistemas de transmissão simplificados onde não são fornecidos canais de retorno de mensagens. Neste serviço as mensagens são enviadas somente em uma direção. A camada de link não espera uma resposta confirmando se a mensagem foi recebida corretamente ou indicando que o pedido será processado.

Na classe S2, SEND/CONFIRM, a estação que transmite a mensagem espera uma resposta da estação que recebe. A estação de recebe a mensagem verifica seu conteúdo. Caso nenhum erro seja detectado e o buffer de recebimento esteja disponível, uma mensagem de confirmação positiva (ACK) é retornada à estação solicitante. Se o buffer de recebimento estiver indisponível ou cheio, uma mensagem de confirmação negativa (NACK) é devolvida. Caso um erro tenha sido detectado na mensagem recebida, não será enviada nenhuma resposta e a mensagem recebida será descartada. Isto ocorre, pois uma vez não recebendo uma mensagem de confirmação, a estação primária (a que envia os pedidos) após a ocorrência de um "timeout" irá retransmitir a mensagem anterior.

A classe S3, REQUEST/RESPOND permite operações de "leitura". A estação primária irá solicitar uma informação à outra estação que por sua vez irá devolver uma mensagem que contenha as informações solicitadas. Caso a estação secundária não disponha dos dados solicitados, uma mensagem de confirmação negativa (NACK) é devolvida. A estação primária irá retransmitir a mensagem caso não receba resposta ou caso esta apresente erros.

II.3.1.4 - Padrões de formato de frames

A seguir definiremos quatro classes distintas de formato de frames, definidas para permitir a compatibilidade com os avanços e mudanças sofridas pelas informações além de garantir as o atendimento às diferentes classes de integridade definidas e demandadas pelos sistemas de telemetria e controle.

Os tipos de formato de frames ilustrados na Tabela 1 podem ser definidos nas classes: FT 1.1, FT 1.2, FT 2 ou FT 3.

A classe FT 1.1 define um bloco com distância Hamming igual a 2. Este bloco é formado por 8 bits de informação, precedidos de 1 bit de início (start bit) e sucedidos por 1 bit de paridade e 1 bit de parada (stop bit).

Seqüências de blocos da classe FT 1.1, acompanhadas por um bloco de checksum formam a classe FT 1.2 com distância de Hamming igual a 4. Esta classe é utilizada na transmissão de mensagens baseadas na norma IEC 870-5-101, na qual se baseia este projeto.

A classe FT 2 é definida por um bloco com distância de Hamming igual a 4 e que contém até 15 bytes complementados por um bloco de verificação de 1 byte.

A classe FT 3 é definida por um bloco com distância de Hamming igual a 6 e que contém até 16 bytes complementados por um bloco de verificação de 2 bytes.

Tanto a classe FT 1.2 como a FT 2 preenchem os requisitos necessários para a integridade do tipo I2. Enquanto de a FT 1.2 garante maior eficiência, a outra garante menor quantidade de erros de transmissão.

Figura II.7 – Formatos de frame

Classe de formato	Distancia de Hamming	Classe de Integridade	Formato do Frame	Especificações
FT 1.1	2	I1	0 P 1 0 P 1 0 P 1 0 P 1 1 8 1 1 1 2 3 i	1 bit início (0) 1 bit fim (1) 1 bit de paridade par (P)
FT 1.2	4	I2	0 P 1 0 P 1 0 P 1 0 CS P 1 1 8 1 1 1 2 i i + 1	1 bit início (0) 1 bit fim (1) 1 bit de paridade par (P) A cada frame: 8 bits de checksum módulo 256 (CS)
FT 2	4	I2	CS-8 CS-8 k=8,16,14 ... 120 8 k=8,16,14 ... 120 8	O byte de CRC é formado pelo polinômio: X7 + X6 + X5 + X2 +1
FT 3	6	I2	CS-16 CS-16 k=8,16,14 ... 128 16 k=8,16,14 ... 128 16	O byte de CRC é formado pelo polinômio: X16 + X13 + X12 + X11 + X10 + X8 + X6 + X5 + X2 + 1

II.3.2 - IEC 870-5-2

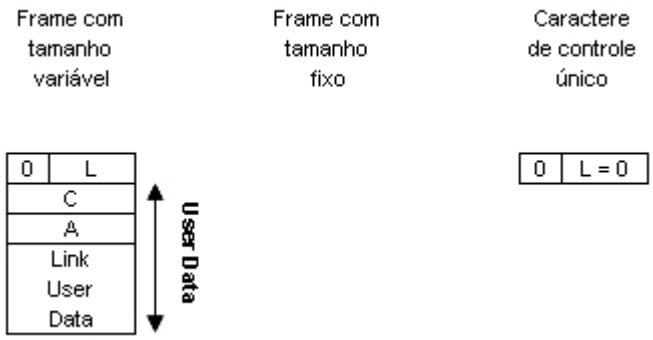
Nesta seção, serão definidos, de forma mais específica, os tipos de informação presentes nas diferentes mensagens enviadas e recebidas. Iremos definir a função de bytes específicos, como o byte de controle, o de endereço, o byte de início e de fim e o byte de tamanho. Também serão discutidas as características de cada tipo de frame de transmissão.

II.3.2.1 – Formatos de frame

Como ilustrado na seção anterior, possuímos quatro tipos de formatos de frame: FT 1.1, FT 1.2, FT 2 e FT 3. A seguir serão detalhados os formatos das mensagens para esses tipos de formatos de frame.

II.3.2.1.1 – Formato FT 1.1

Figura II.8 – Formato de frame FT 1.1

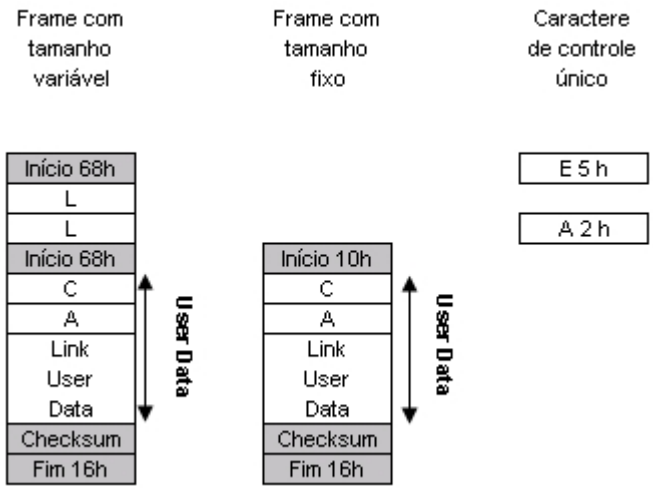


Nesta figura, L representa o tamanho da mensagem menos 1 byte. De outra forma podemos dizer que L é o tamanho do "user data", que é igual ao tamanho do "link user data" mais 2 bytes (o byte de controle e o byte de endereço). C representa o byte de controle e A o byte de endereço. Ambos serão detalhados posteriormente.

O formato FT 1.1 não possui um tipo de mensagem com tamanho fixo. O tipo de tamanho variável é usado em ambos os casos. O caractere único, também chamado de caractere único de controle normalmente é utilizado quando queremos dar respostas rápidas de confirmação do tipo ACK ou NACK.

II.3.2.1.2 - Formato FT 1.2

Figura II.9 – Formato de frame FT 1.2



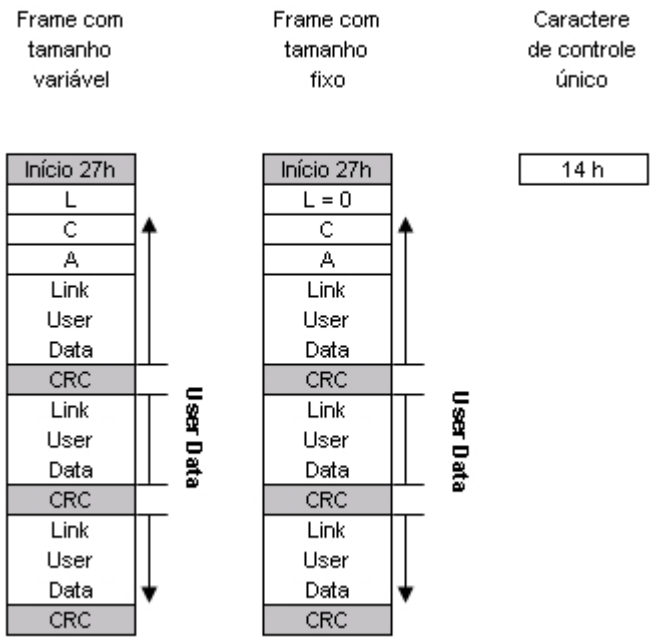
Este é o formato utilizado pelo protocolo baseado na norma IEC 870-5-101. Os três tipos de estruturas - a mensagem de tamanho variável, a de tamanho fixo e o caractere único de controle - estão bem definidas neste formato de frame de mensagem.

Na estrutura de tamanho variável, o primeiro e o quarto byte são classificados como bytes de início de frame e para este formato específico possuem o valor 68 em hexadecimal. O segundo e terceiro byte indicam o tamanho da parte variável da mensagem, ou seja, o tamanho do "user data". O quinto e sexto byte representam o byte de controle e o de endereço respectivamente. O penúltimo byte é o byte de verificação ou checksum que é calculado pela soma módulo 256 de cada um dos valores dos bytes do "user data". Por fim, o último byte representa o caractere de termino de frame e para este formato é definido como sendo o valor 16 em hexadecimal.

A estrutura de tamanho fixo apresenta um byte de início como o valor 10 em hexadecimal, seguido pelos bytes de controle e endereço. A norma permite que seja definido de comum acordo entre as partes, segundo a implementação adotada, um número de bytes para o "user data", porém normalmente são utilizados apenas os bytes de controle e de endereço, ou seja, o "link user data" normalmente não é utilizado. Da mesma forma que a estrutura de tamanho variável, a mensagem termina com um byte de verificação e outro de término, sendo que este também possui o valor 16 em hexadecimal. O caractere único está definido e assume o valor E5 em hexadecimal.

II.3.2.1.3 - Formato FT 2

Figura II.10 – Formato de frame FT 2



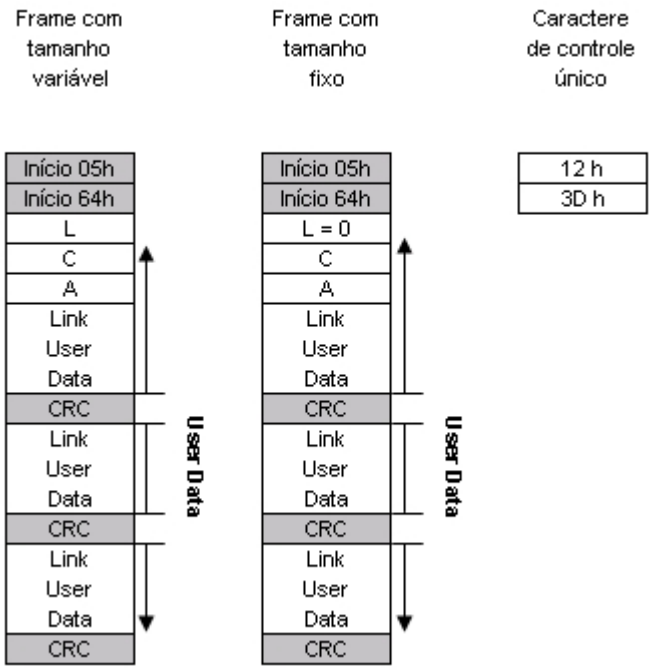
Como podemos verificar, este formato apresenta basicamente os mesmos elementos que o anterior, como os bytes de tamanho, endereço, controle e início com algumas diferenças. O byte de fim de mensagem não está presente. Tanto para a mensagem de tamanho fixo como para a de tamanho variável o byte de início é igual e a diferença entre as mensagens será dada pelo valor do byte de tamanho. Conforme descrito para o formato FT 1.2, o tamanho do "user data" é definido de acordo com a implementação adotada.

Os aspectos fundamentais deste formato estão na presença de um tipo diferente de byte de verificação e no fato de que sua presença poder ocorrer diversas vezes ao longo da mensagem. Segundo especifica a norma, um conjunto de até 15 bytes do campo "user data" é sucedido por um byte de verificação, desta forma caso os bytes que compõem o "user data" sejam maiores do que 15 bytes, estes serão divididos em blocos distintos, cada um completado por um byte de verificação. O byte de verificação por sua vez é formado segundo padrão CRC (Cyclic Redundancy Check). O byte de CRC forma um código que é gerado pelo polinômio $X^7 + X^6 + X^5 + X^2 + 1$, completado por um bit de paridade par referente a todos os bits do bloco. Após serem gerados, os 8 bits são invertidos.

O caractere único é definido e apresenta o valor 14 em hexadecimal.

II.3.2.1.4 - Formato FT 3

Figura II.11 – Formato de frame FT 3



O último formato definido pela norma é basicamente igual ao formato definido anteriormente, porém três diferenças devem ser destacadas. Primeiro, podemos notar a presença de dois bytes de início, sendo o primeiro de valor 5 e o segundo igual a 64, ambos em hexadecimal. Segundo, os bytes de CRC são precedidos de blocos de 16 bytes, e não de 15 como foi definido na seção anterior. Por último, o campo de CRC possui 2 bytes e não apenas 1, além do que o polinômio de formação é igual a $X^{16} + X^{13} + X^{12} + X^{11} + X^{10} + X^{8} + X^{6} + X^{5} + X^{2} + 1$. Os 16 bits gerados são invertidos.

II.3.2.2 – Tipos de Transmissão de Mensagens

As mensagens podem ser classificadas quanto ao tipo de transmissão em duas classes: balanceado e não-balanceado. Caracterizamos como não-balanceado um sistema de transmissão do tipo cliente-servidor, onde uma das pontas é responsável por enviar pedidos ou comandos enquanto que a outra se limita apenas a responder as solicitações recebidas. Em um sistema balanceado, ambas as pontas podem iniciar a transmissão de dados, sendo assim, normalmente neste sistema é necessário que existam dois canais de comunicação (um em cada direção), pois mensagens podem estar sendo originadas simultaneamente em ambas as pontas, enquanto que em um sistema não-balanceado podemos utilizar apenas um canal de transmissão visto que a remota não pode originar mensagem sem que estas tenham sido solicitadas previamente pela estação primária. Como este trabalho irá abordar o modo de transmissão não-balanceado não iremos detalhar as características do modo de transmissão balanceado.

II.3.2.3 – Byte de Controle

O byte de controle contém informações de caracterizam a direção da mensagem, o tipo do serviço utilizado e dá suporte a funções de controle para suprimir perdas ou mensagens duplicadas.

Figura II.12 – Byte de controle

A seguir são explicados os bits que compõem o byte de controle:

- RES: bit reservado. Não é utilizado;
- FCB: frame count bit. Este bit alterna entre 0 e 1 para sucessivas mensagens de SEND/CONFIRM ou REQUEST/RESPOND por estação remota. O FCB é utilizado para deletar perdas ou mensagens duplicadas na transferência de dados. A estação primária alterna o FCB para cada nova mensagem de SEND/CONFIRM ou REQUEST/RESPOND transmitida para a mesma estação secundária. Desta forma, a estação primária mantém uma cópia do FCB por estação remota. Se ocorreu timeout em uma resposta esperada ou esta foi recebida incorretamente, a mesma mensagem de SEND/CONFIRM ou REQUEST/RESPOND é enviada novamente com o mesmo valor do FCB da última mensagem enviada. Desta forma a estação secundária saberá que houve um erro na transmissão e a resposta à mensagem enviada será transmitida novamente. No caso de comandos de "reset", o FCB será sempre 0 e uma vez recebido este comando, a estação secundária estará sempre pronta para esperar a próxima mensagem enviada pela estação primária com o bit FCV (explicado a seguir) e FCB iguais a 1;
- FCV: frame count bit valid. Indica se a alternância de valores para o FCB é valida, ou seja, caso FCV seja igual a 1 e a n-ésima mensagem enviada da estação primária para a remota apresente FCB igual a 0, a próxima mensagem enviada apresentará o FCB igual a 1, levando em consideração que a mensagem anterior foi enviada e sua resposta correspondente recebida corretamente. Serviços de SEND/NO REPLY, mensagens de broadcast, e outros serviços de transmissão que ignoram a perda de mensagens ou mensagens duplicadas não alternam o FCB e indicam isto através do FCV igual a 0;

- DFC: data flow control. Se igual a 1 indica que futuras mensagens enviadas à estação secundária poderão causar overflow, caso contrário informa que próximas mensagens recebidas serão aceitas;
- ACD: access demand. Existem duas classes de mensagens de dados: classe 1 e 2. Caso ACD seja igual a 1, a estação remota indicará à primária que existe a necessidade a aquisição de dados de classe 1. Caso contrário, apenas estão presentes dados de classe 2 na remota, não sendo necessária a transmissão em classe 1. A diferença entre esses dois tipos de mensagens será explicada em maiores detalhes posteriormente;
- PRM: primary message. Caso a mensagem tenha sido enviada pela estação primária para a estação secundária este bit assume o valor 1. Se a mensagem tiver sido enviada pela remota para a estação primária assume o valor 0;
- Função: indica o tipo de mensagem transmitida segundo tabelas abaixo.

Tabela II.1 – Funções de código para transmissão não-balanceada de mensagens enviadas pela estação de controle

Código da Função	Tipo do frame	Função de serviço	FCV
0	SEND / CONFIRM	Reset do link remoto	0
1	SEND / CONFIRM	Reset do processo do usuário	0
2	SEND / CONFIRM	Reservado para transmissão balanceada	-
3	SEND / CONFIRM	Dados	1
4	SEND / NO REPLY	Dados	0
5		Reservado	-
6-7		Reservado	-
8	REQUEST por acesso sob demanda	Resposta determina acesso sob demanda	0
9	REQUEST / RESPOND	Pedido por status do link	0
10	REQUEST / RESPOND	Pedido por dados de classe 1	1
11	REQUEST / RESPOND	Pedido por dados de classe 2	1
12-13		Reservado	-
14-15		Reservado	-

Tabela II.2 – Funções de código para transmissão não-balanceada de mensagens enviadas pela remota

Código da Função	Tipo do frame	Função de serviço
0	CONFIRM	ACK: confirmação positiva
1	CONFIRM	NACK: mensagem não aceita ou link ocupado
2-5		Reservado
6-7		Reservado
8	RESPOND	Dados
9	RESPOND	NACK: dado requisitado não disponível
10		Reservado
11	RESPOND	Status do link ou acesso sob demanda
12		Reservado
13		Reservado
14	-	Serviço de link não está funcionando
15	-	Serviço de link não implementado

II.3.2.4 – Classes de mensagens de dados

Segundo citado anteriormente, pode-se classificar as mensagens de aquisição de dados em mensagens de classe 1 ou 2. As mensagens de classe 1 são definidas como mensagens de alta prioridade ou mensagens geradas por eventos, enquanto que as de classe 2 são definidas como sendo de baixa prioridade ou transmissões cíclicas.

Na remota, à medida que as mensagens são geradas elas são colocadas em uma fila ou buffer para serem transmitidas para a estação primária. A primária por sua vez solicita as mensagens da remota de acordo com a ordem em que elas foram colocadas na fila. Porém, existem certos tipos de mensagens que precisam ser transmitidas rapidamente, desta forma, as mensagens de dados são definidas em duas classes. Caso existam mensagens de alta prioridade (classe 1) esperando para serem transmitidas da remota para a primária, na sua próxima mensagem de resposta a remota irá indicar que existem mensagens de classe 1, sinalizando o bit ACD igual a 1. Na próxima mensagem de aquisição de dados que a primária fizer à remota, a primária

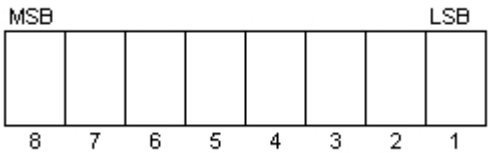
pedirá por mensagens de classe 1. Este processo se repete até que não existam mais mensagens de classe 1 na remota ou caso esta decida não enviar mais mensagens deste tipo. Em ambos os casos a remota indica o ACD como 0, assim na próxima mensagem a primária solicitará mensagens de classe 2.

Não existe na norma nenhuma definição explícita sobre quais tipos de mensagens devem ser classificadas com classe 1 ou 2, ficando esta decisão sob responsabilidade dos responsáveis pelo desenvolvimento.

II.3.2.5 – Byte de Endereço

O byte de endereço especifica o endereço da estação. Este campo é parte integrante dos frames das mensagens como pode ser visto anteriormente. Quando é transmitido das estações que iniciam o serviço de transmissão de dados (estações primárias) para as estações receptoras (estações secundárias) este campo indica a estação de destino. Quando transmitido da remota para a estação primária, indica o endereço de origem.

Figura II.13 – Byte de endereço



O número de bytes deste campo pode variar de acordo com a implementação e a norma diz que este valor deve ser acordado entre fabricante e usuário.

II.3.3 – IEC 870-5-3

Esta seção especifica regras para a estruturação de elementos de dados da camada da aplicação, nos frames das mensagens, em sistemas de telemetria e controle. Estas regras são apresentadas como padrões genéricos que podem ser usados para dar suporte a uma grande variedade de aplicações presentes e futuras. O layout é projetado para minimizar o overhead organizacional de aplicações de aquisição de dados e tarefas de controle supervisionado a um mínimo necessário com possíveis extensões para tarefas especiais. Desta forma, é possível admitir escolhas específicas da aplicação ou do sistema para visualização de dados, estruturas de endereços e mecanismos de agregação de objetos de informação em um frame.

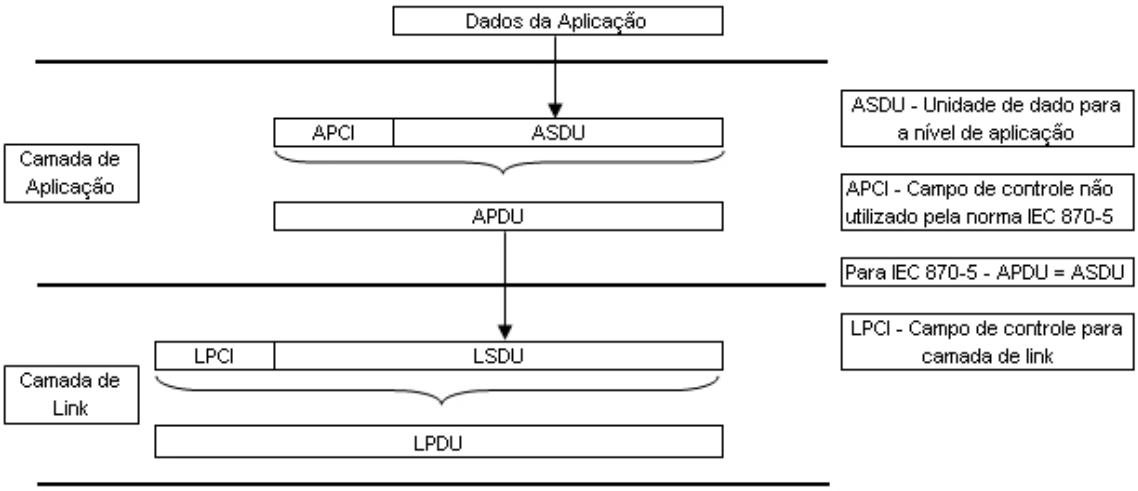
Esta seção descreve a estrutura geral dos dados no nível de aplicação, sem todavia especificar detalhes sobre campos de informação e seus conteúdos. São descritas apenas regras gerais e básicas para futuras especificações de unidades de dados.

Conforme já citado anteriormente, para a especificação do protocolo baseado na norma IEC 870-5-101 será utilizado o modelo EPA (Enhanced performance architecture) que possui apenas 3 camadas (física, link e aplicação). Este modelo é uma especialização do modelo OSI que possui 7 camadas. Para sistemas de controle e telemetria que requerem uma alta velocidade de resposta em redes com banda limitada de transmissão, este modelo se mostra mais adequado devido à redução oferecida no tempo de resposta.

II.3.3.1 – Estrutura de Dados de Aplicação

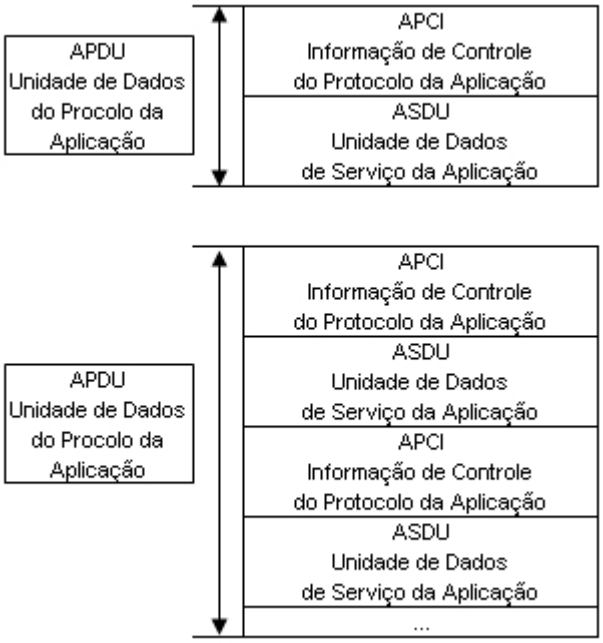
As relações entre os elementos de dados para o modelo de referência EPA pode ser ilustrado abaixo:

Figura II.14 – Relação entre unidades de dados



As estruturas genéricas das unidades de dados do protocolo da aplicação (APDU) usados em aplicações de telemetria são:

Figura II.15 – Estrutura genérica da unidade de dados do protocolo da aplicação (APDU)

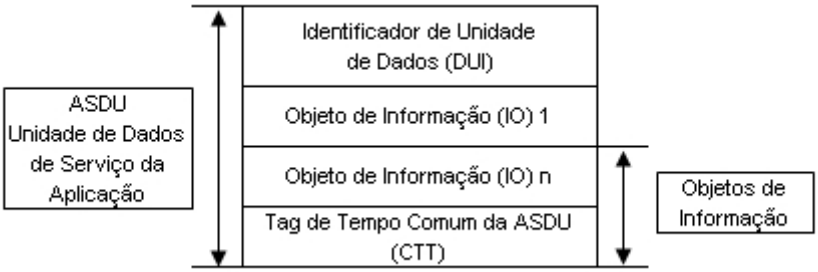


Um frame pode conter um ou mais conjuntos de APCIs e ASDUs como mostrado acima.

II.3.3.1.1 – Unidade de dados de serviço da aplicação (ASDU)

A unidade de dados de serviço da aplicação (ASDU) consiste de um identificador de unidade de dados (DUI) e um ou mais objetos de informação (IO). A estrutura genérica de uma ASDU é:

Figura II.16 – Estrutura genérica da unidade de dados de serviço da aplicação (ASDU)



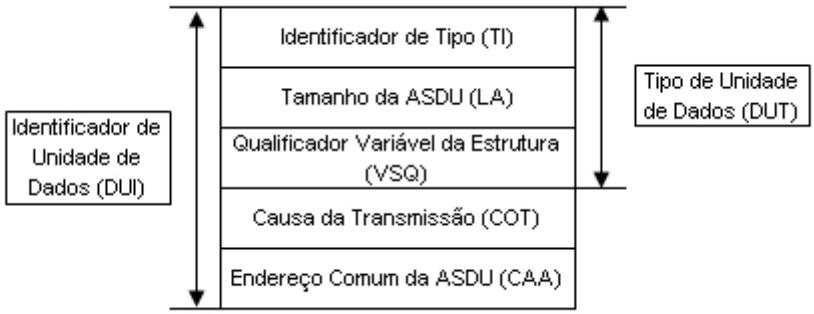
O tag de tempo comum da ASDU (CTT) pode estar localizado como o último endereço do objeto de informação (IOA). A presença do CTT é definida no identificador de unidade de dados.

II.3.3.1.2 – Identificador de unidade de dados

O DUI consiste de um identificador de tipo (TI), um tamanho da ASDU (LA), um qualificador variável da estrutura (VSQ), uma causa da transmissão (COT) e um endereço comum da ASDU (CAA) sendo que os quatro últimos campos são opcionais.

A combinação dos três primeiros elementos citados anteriormente é chamada de tipo de unidade de dados (DUT).

Figura II.17 – Estrutura genérica do identificador de unidade de dados (DUI)



O identificador de tipo é um código que identifica inequivocamente o tipo da ASDU dentro da coleção de possíveis tipos para uma determinada configuração ou sistema escolhido. Caso presente, o tamanho da ASDU indica o tamanho total desta em bytes. Da mesma forma, o qualificador variável da estrutura indica variações desta para ASDUs específicas, que podem variar para diferentes instâncias de uma comunicação de dados. O identificador de tipo permite que a aplicação na estação receptora encaminhe os dados para os processos adequados, de acordo com o tipo de dados. Permite também que o processo possa visualizar o tipo de informação contida no conjunto de dados, para que estes possam ser alocados dentro de estruturas específicas. Se o identificador de unidade de dados estiver presente, o identificador de tipo é o único campo não opcional.

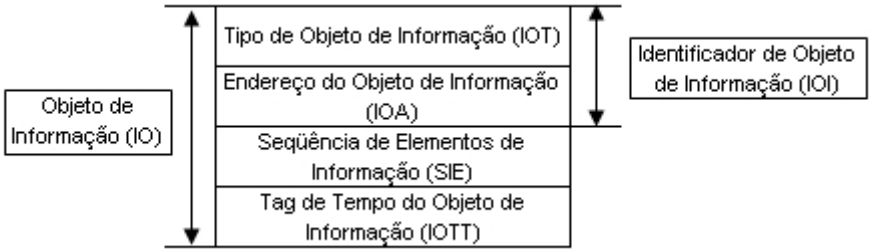
A causa da transmissão também pode ser incluída no DUT caso não tenha sido definida explicitamente.

Se o endereço comum da ASDU estiver presente, este estará sempre localizado antes dos objetos de informação.

II.3.3.1.3 – Objetos de informação

A ASDU pode incluir um ou mais objetos de informação. A estrutura genérica dos objetos de informação é:

Figura II.18 – Estrutura genérica do objeto de informação (IO)



Um IO pode conter um identificador de objeto de informação (IOI), assim como seqüências de elementos de informação. Um IOI pode incluir o tipo de objeto de informação e o endereço do mesmo.

Um tipo de objeto de informação (IOT) pode ser definido caso haja diferentes estruturas de objetos que não estejam definidas no tipo de unidades de dados. O endereço do objeto de informação, assim como a seqüência de elementos de informação (SIE) serão explicados com mais detalhes posteriormente.

Cada objeto de informação pode opcionalmente ser complementado com um tag de tempo. Se este campo for utilizado, então sempre será inserido no final do IO.

A tabela abaixo procura ilustrar de forma mais global como estão organizados os diversos elementos que compõem as ASDUs e que foram mostrados nas figuras anteriores. Campos opcionais podem ser omitidos, visto que não é obrigatório implementar toda estrutura da ASDU.

Tabela II.3 – Informações contidas em uma ASDU

Identificador de Unidade de Dados	Tipo de Unidade de Dados	Identificador de Tipo	Tipo da ASDU
			Estrutura do Identificador de Unidade de Dados
			Tipo de Objetos de Informação: único elemento, combinação de elementos
			Descrição do elemento de informação: tipo, formato
			Estrutura de informação de tempo
			Tag de tempo comum ou individual para todos os objetos
			Detalhes no esquema de endereçamento
		Tamanho da ASDU	Número de bytes da ASDU
		Qualificador Variável da Estrutura	Número de Objetos de Informação
			Número de elementos de informação na Seqüência de Objetos de Informação
			Alguns tipos de objetos
	Causa da Transmissão		Periódico, espontâneo, inicialização, etc.
	Endereço Comum da ASDU		Endereço associado a todos os objetos da ASDU
Objeto de Informação	Tipo de Objeto de Informação		
	Endereço do Objeto de Informação		
	Seqüência de Elementos de Informação		
	Tag de Tempo do Objeto de Informação		
Objeto de Informação	Tag de Tempo Comum da ASDU		

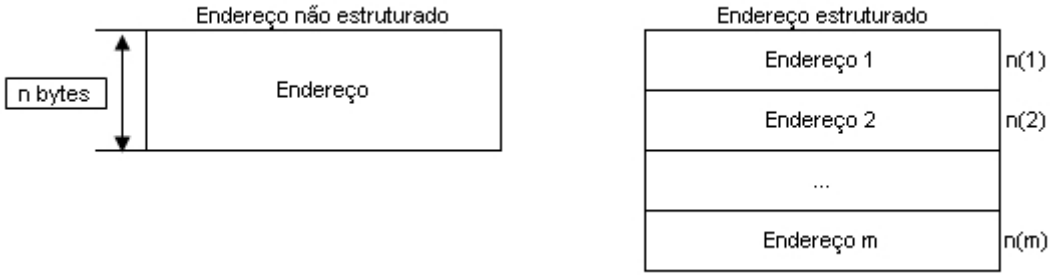
II.3.3.1.4 - Identificação dos objetos de informação

Em sistemas de telemetria, de forma a dar suporte a uma grande variedade de possíveis configurações é necessário que os objetos de informação sejam identificados. Em sistemas simples, estes podem ser identificados por seus endereços físicos. Os endereços são normalmente estruturados para representar a imagem dos processos controlados.

Endereços não-estruturados são normalmente usados para distinguir entre diferentes objetos de informação levando em consideração números selecionados a partir de um conjunto único

de valores. Já os endereços estruturados levam em consideração estruturas tecnológicas, físicas, topológicas ou geográficas. Para este esquema, é importante se definir um número muito grande de endereços por estágio, para permitir a expansão de um sem que haja a sobreposição de outro.

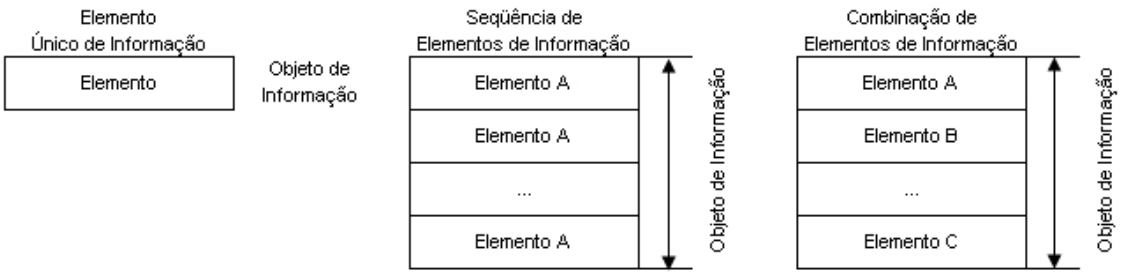
Figura II.19 – Dois tipos de endereçamento do objeto de informação (IO)



II.3.3.1.5 – Seqüências de elementos de informação

Podemos distinguir três tipos de seqüências de elementos de informação, segundo ilustrado abaixo:

Figura II.20 – Possíveis tipos de elementos de informação (IE)



No primeiro caso a seqüência consiste em um único elemento de informação que é identificado pela associação do endereço do objeto de informação com o endereço comum da ASDU. Como exemplos de elementos únicos de informação podemos citar: comandos, eventos, valores de estado e valores analógicos.

No caso da seqüência de elementos de informação engloba-se uma lista bem definida de elementos, como valores de medidas do mesmo tipo. Neste caso, o IOA e o CAA caracterizam o endereço associado ao primeiro elemento da seqüência, enquanto que os elementos seguintes são identificados de acordo com uma seqüência de endereçamento pré-definida.

Para o caso da combinação de elementos de informação, a seqüência de elementos é composta por uma lista bem distinta destes, como por exemplo, na combinação de valores analógicos e digitais caracterizando o estado de uma linha de transmissão. Desta forma, o IOA e o CAA especificam o endereço associado a todo o objeto de informação, e os elementos individuais são definidos de acordo com algum modelo estrutural pré-definido.

Os elementos são quantidades variáveis representados por códigos e tipos de dados pré-definidos. Estas quantidades ou valores são do tipo: booleano, inteiro, real, entre outros.

II.3.4. IEC 870-5-4

Nesta seção, a norma estabelece regras para definir os elementos de informação além de apresentar um conjunto desses elementos, em particular variáveis de processo analógicas e digitais que são freqüentemente utilizadas em procedimentos de telemetria

São apresentadas regras para se definir os elementos de informação. Estas regras compreendem métodos para declarações semânticas, ou seja, a atribuição de interpretação funcional aos campos de informação definidos. São definidos também os tipos básicos de dados (inteiro, real, etc.) segundo os critérios definidos para os elementos de informação além de serem definidos também tipos de elementos de informação importantes e muito utilizados nas aplicações de controle e telemetria.

Visto que o foco desta seção está centrado em questões muito específicas, como o formato dos tipos de dados utilizados pelos elementos de informação e de forma a manter a linha de pensamento apresentada, as páginas relevantes presentes na norma serão citadas em anexo a este documento para permitir futuras consultas porém não serão descritas nesta seção.

II.3.5. IEC 870-5-5

Nesta seção serão descritas diversas funções básicas que executam procedimentos normalmente utilizados em sistemas de telemetria. As aplicações descritas nesta seção residem além da sétima camada do modelo OSI e utilizam serviços da camada de aplicação para a transmissão de mensagens. As funções que serão apresentadas adiante são ilustradas através de diagramas que indicam a seqüência de mensagens trocadas entre a estação primária e a remota. As primeiras funções mostradas, a de inicialização da estação e de aquisição de dados através de pooling, representam a base para a execução de futuras aplicações básicas.

Os procedimentos seqüenciais de transmissão são descritos por flechas. Cada flecha representa uma ASDU ou mensagem. Uma estrutura de nomes é usada para classificar e

identificar os diferentes tipos de mensagens, lista essa que pode ser completada por diferentes normas futuras.

Tabela II.4 – Tipos de ASDU

Tipo de informação - nível 1

Informação de monitoração	M
Informação de controle	C
Parametro	P
Transferência de arquivo	F

Tipo de informação - nível 2

Informação de monitoração	M
Informação de estado simples	M_SP
Informação de estado duplo	M_DP
Medidas	M_ME
Eventos de proteção	M_EP
Valores totalizados	M_IT
Informação de posição	M_ST
Strings de bits e bytes	M_BO
Fim de inicialização	M_EI
Disponibilidade da camada de aplicação	M_AA

Informação de controle	C
Comando simples	C_SC
Comando duplo	C_DC
Comando de set-point	C_SE
Comando de regularização de posição	C_RC
Comando de interrogação	C_IC
Comando de sincronismo de clock	C_CS
Aquisição de atrazo	C_CD
Comando de contra-interrogação	C_CI
Comando de teste	C_TS
Comando de reset de processo	C_RP
Comando de leitura	C_RD
Fim de inicialização	C_EI

Parametro	P
Parametro para medidas	P_ME
Ativação de parametro	P_AC

Transferência de arquivo	F
Diretório	F_DR
Seleção ou chamada de arquivo, seção ou diretório	F_SC
Última seção ou segmento	F_LS
ACK para arquivo ou seção	F_AF
Arquivo pronto	F_FR
Seção pronta	F_SR
Segmento	F_SG

Os rótulos das ASDU apresentadas na tabela anterior seguem uma ordem hierárquica para melhor identificar os tipos de mensagem dentro de um grupo, além de permitir a especificação de rótulos mais específicos em outras normas.

Um terceiro nível é usado por diferentes normas e define o tipo específico da ASDU, o uso de um time tag, etc. A primeira letra do terceiro nível hierárquico especifica a presença de um time tag (N significa ausência de time tag e T significa presença de time tag), a segunda letra define o tipo. Cada norma pode especificar os seus tipos em ordem alfabética começando com a letra A. Por exemplo:

Medida, valor normalizado sem tag de tempo (tipo A)	M_ME_NA
Medida, valor em escala com tag de tempo (tipo B)	M_ME_TB
Comando simples, tipo A sem tag de tempo	C_SC_NA

Complementando, um número final é adicionado para indicar que norma está definindo a ASDU. Por exemplo:

Norma 101	M_ME_NA_1 ou C_SC_NA_1
Norma 102	M_ME_NA_2 ou C_SC_NA_2

Este sistema de rótulos é aberto e pode ser completado, caso necessário, em todos os níveis hierárquicos por diferentes normas.

A seguir serão descritos os tipos mais comuns de mensagens trocadas entre as estações, o ciclo destas mensagens e explicações sobre a funcionalidade destas.

II.3.5.1 – Inicialização da estação

Os procedimentos de inicialização de estações são requeridos de forma a configurar as estações em corretos estados de operação antes do início das operações de telemetria dependentes das camadas de aplicação. É necessário fazer a distinção entre um procedimento de cold-start e de warm-start. Um cold-start é uma inicialização na qual as informações contidas nas variáveis de processo são apagadas antes da atualização no banco de dados dos estados atuais. Um warm-start é um procedimento de reinicialização onde os valores contidos nas variáveis de estados não são apagados

As estações de controle normalmente possuem equipamentos de controle e banco de dados redundantes que garantem a integridade com relação a possíveis falhas sem a perda de informações. Neste caso, não é necessário dar início a um procedimento de atualização dos dados da base de dados da estação de controle. Todavia, após um procedimento de ativação de uma estação de controle ou uma reinicialização total do sistema, é necessário que sejam trocadas mensagens com o objetivo de atualizar a base da estação primária ou até mensagens de sincronismo de clock. Estas mensagens são chamadas de “general interrogation” e “clock synchronization” respectivamente.

Uma estação secundária pode ser reinicializada através de um comando local assim como através de uma mensagem enviada pela estação de controle.

II.3.5.1.1 – Inicialização da estação de controle

Após a inicialização interna da estação de controle, a camada de link estabelece conexões com as estações remotas. Quando a estação primária está pronta para receber informações de suas remotas, a primeira pode enviar um C_EI (fim da inicialização) para as remotas conectadas (esta mensagem é opcional). Após o recebimento do C_EI, a remota poderá enviar informações de processo para as estações de controle. Estas por sua vez podem dar continuidade ao processo enviado comandos de "general interrogation" e "clock synchronization".

II.3.5.1.2 – Inicialização da estação remota

Caso seja necessário, após a inicialização interna da remota, a camada de link estabelece a conexão com a estação de controle. Caso a remota esteja pronta para enviar informações para a estação de controle, a remota pode enviar um M_EI para a outra estação (esta mensagem também é opcional). Após o recebimento desta mensagem a estação de controle pode dar continuidade aos procedimentos de inicialização.

Figura II.21 – Procedimento geral de inicialização da estação de controle

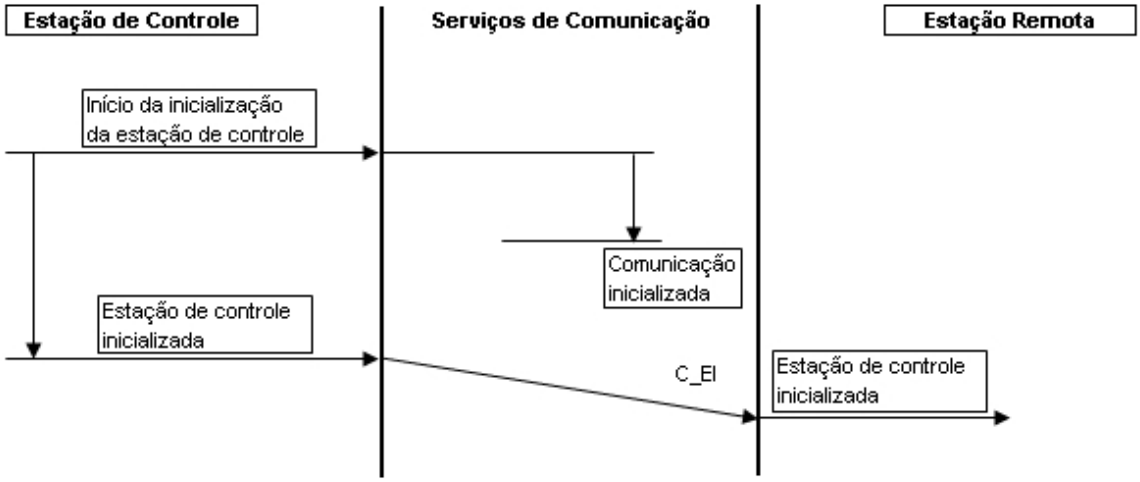
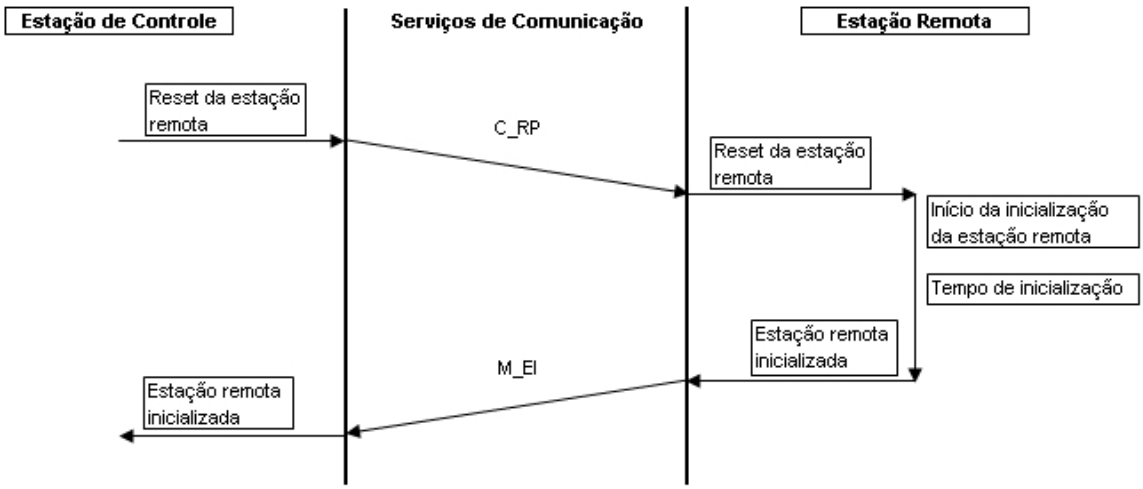


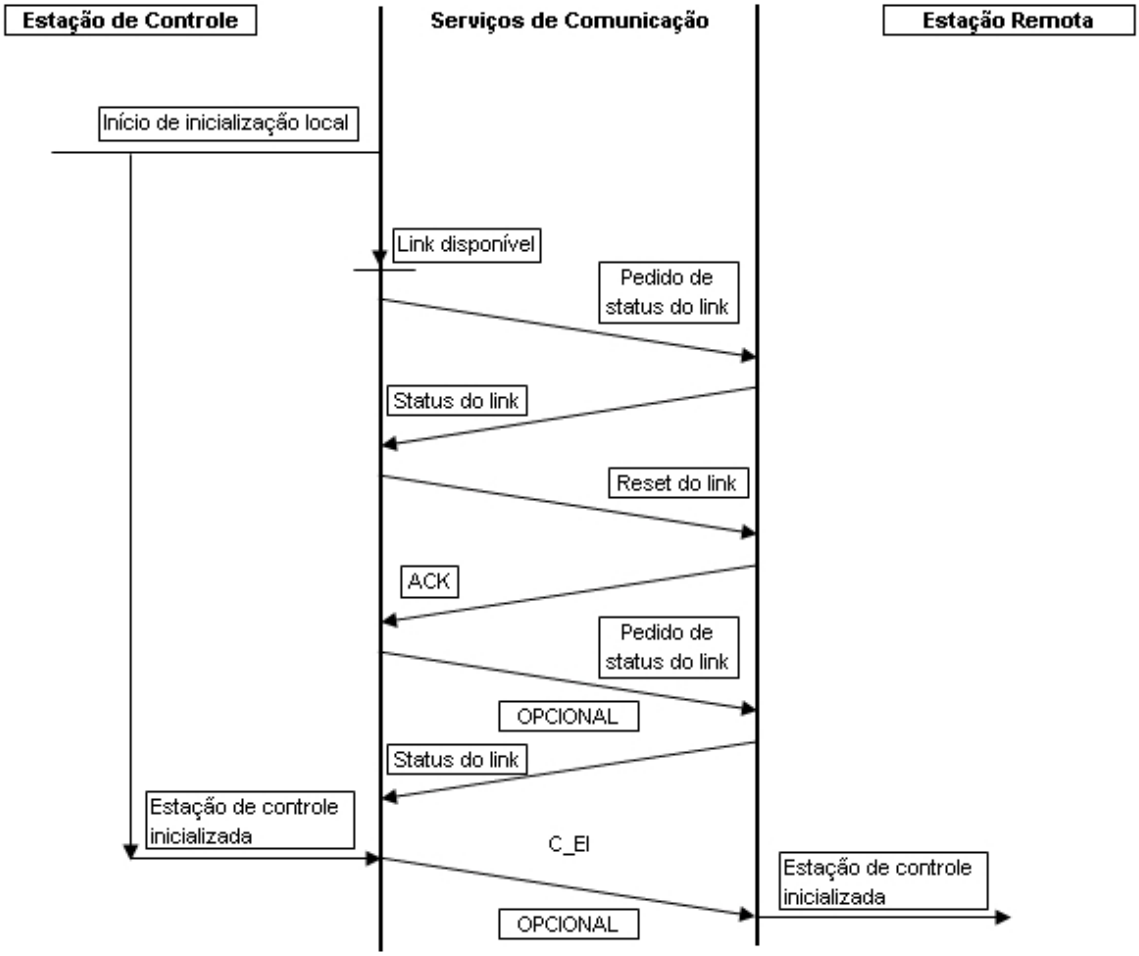
Figura II.22 – Procedimento geral de inicialização da estação remota



II.3.5.2 – Inicialização local da estação de controle em sistemas de transmissão não-balanceados

Após um pedido de inicialização local da estação de controle, a camada de link desta se tornará indisponível e caso houvesse pendente alguma resposta a um pedido enviado, este não poderá ser recebido no momento, apenas após o processo de inicialização. Após a inicialização local, a camada de link é normalmente resetada e habilitada antes de outras funções internas da estação de controle. A camada de link da estação de controle então irá estabelecer a conexão com a camada de link da remota, enviando uma mensagem de "request status of link" que será respondida pela remota com a mensagem "status of link". Em seguida a estação de controle envia uma mensagem de "reset of remote link" que a remota por sua vez responde com um ACK. Este ACK confirma a condição de inicialização da camada de link da remota. Após a completa inicialização das funções de aplicação, uma mensagem de C_EI pode ser enviada indicando que a camada de aplicação está pronta para transmitir mensagens. Após a inicialização, a estação de controle atualiza as informações dos estados das variáveis da remota através de procedimentos "general interrogation" e "clock synchronization".

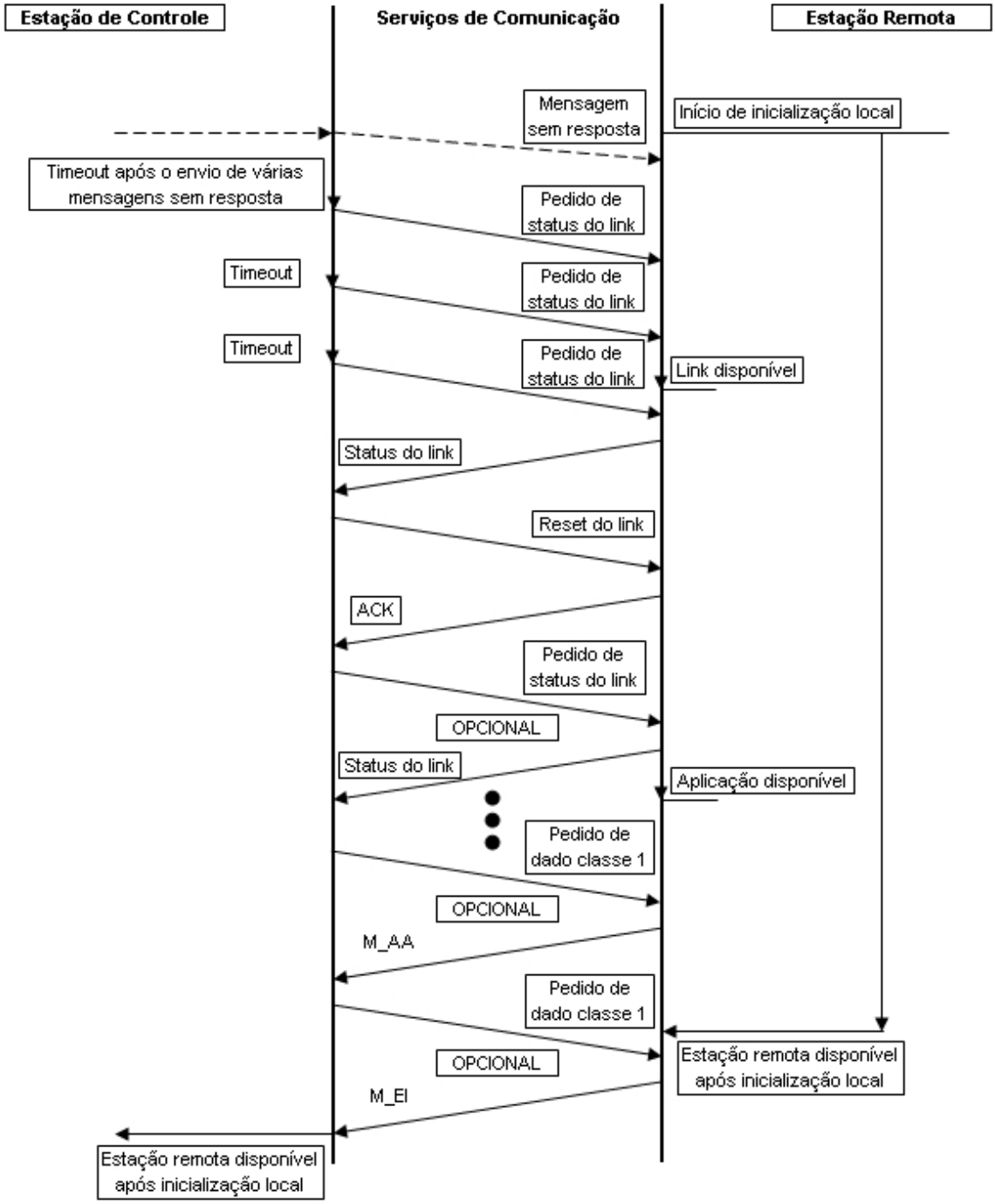
Figura II.23 – Inicialização local da estação de controle



II.3.5.3 – Inicialização local da estação remota em sistemas de transmissão não-balanceados

Após a inicialização da remota, a estação de controle determina de a camada de link da remota está desconectada devido a mensagens enviadas que não obtiveram respostas. Após um certo número de repetições sem sucesso, a estação de controle procura estabelecer conexão com a camada de link da remota, transmitindo repetidas mensagens de pedido de status do link em específicos intervalos de tempo. Quando a camada de link da remota se torna disponível, esta responde com uma mensagem de status do link. A estação de controle então envia um pedido de reset de link que a remota responde com ACK. Os procedimentos a seguir são opcionais podendo ou não ser implementados. A estação de controle pode então enviar seguindos pedidos de status do link à remota. Quando estes forem respondidos com o status do link, indicando que informações de classe 1 estão disponíveis, a estação de controle irá enviar um pedido de dados de classe 1 podendo ser confirmado por um M_AA (camada de aplicação disponível) ou M_EI (fim da inicialização). A inicialização completa da camada de aplicação da remota pode ser sinalizada com um M_EI. Em seguida, a estação de controle executa os procedimentos de “general interrogation” ou “clock synchronization”.

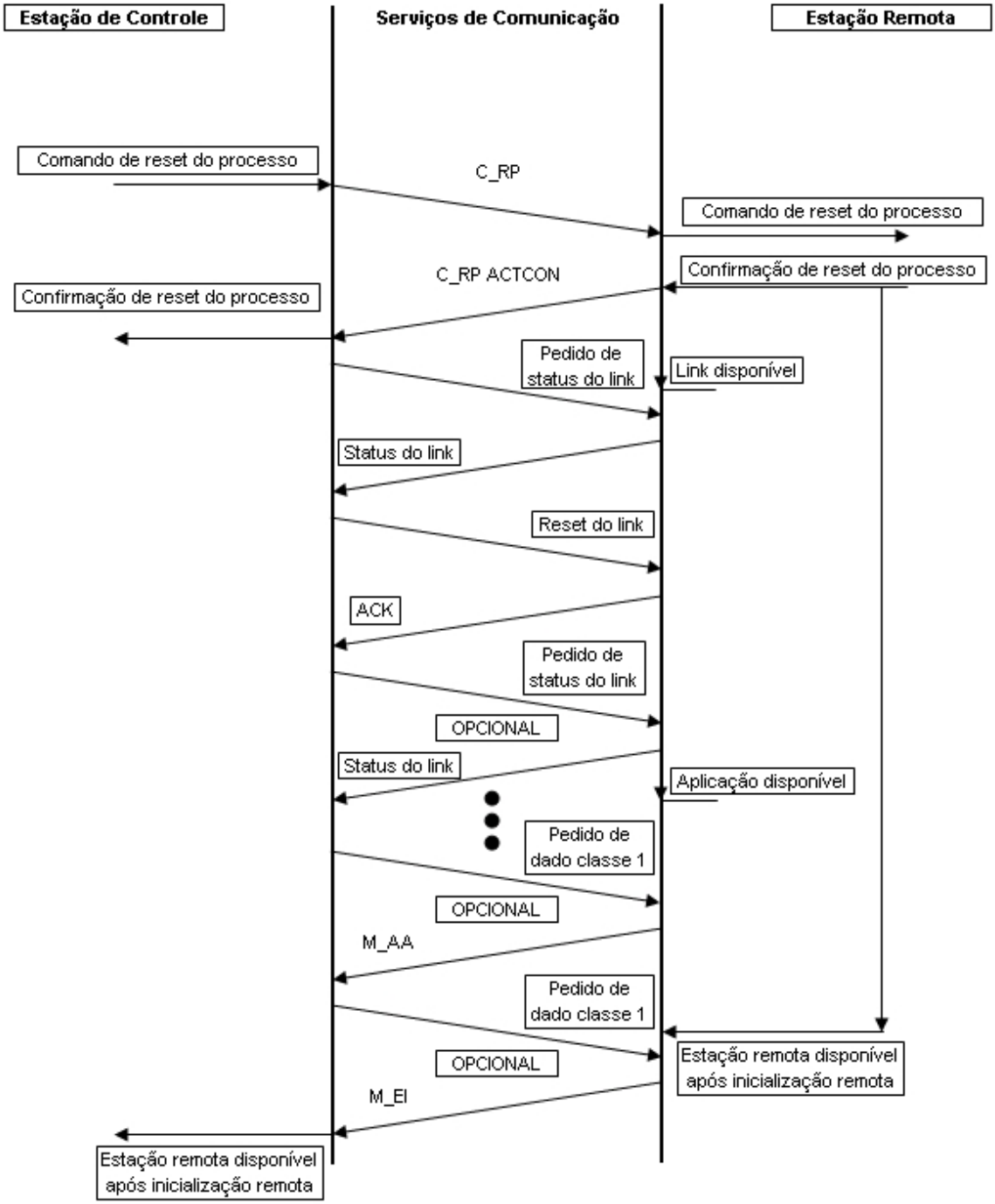
Figura II.24 – Inicialização local da estação remota



II.3.5.4 – Inicialização remota da estação remota em sistemas de transmissão não-balanceados

Após o recebimento de um comando para resetar o processo (C_RP ACT), a estação remota responde com uma mensagem de confirmação (C_RP ACTCON). Em seguida, todos os processos acima da sétima camada, ou seja, da camada de aplicação, são resetados e inicializados. Quaisquer mensagens pendentes de serem enviadas serão descartadas. A partir deste momento, a estação de controle irá executar os procedimentos básicos descritos nos itens anteriores para verificar a disponibilidade da camada de link e de aplicação da remota.

Figura II.25 – Inicialização remota de estação remota



Uma vez descritos os procedimentos adotados pelos diferentes métodos de inicialização, tanto por parte da estação de controle quanto por parte da remota, iremos agora apresentar os

procedimentos de aquisição de dados, responsáveis por coletar as informações presentes nas remotas para que estas possam ser analisadas e tratadas.

II.3.5.5 - Aquisição de dados por "polling"

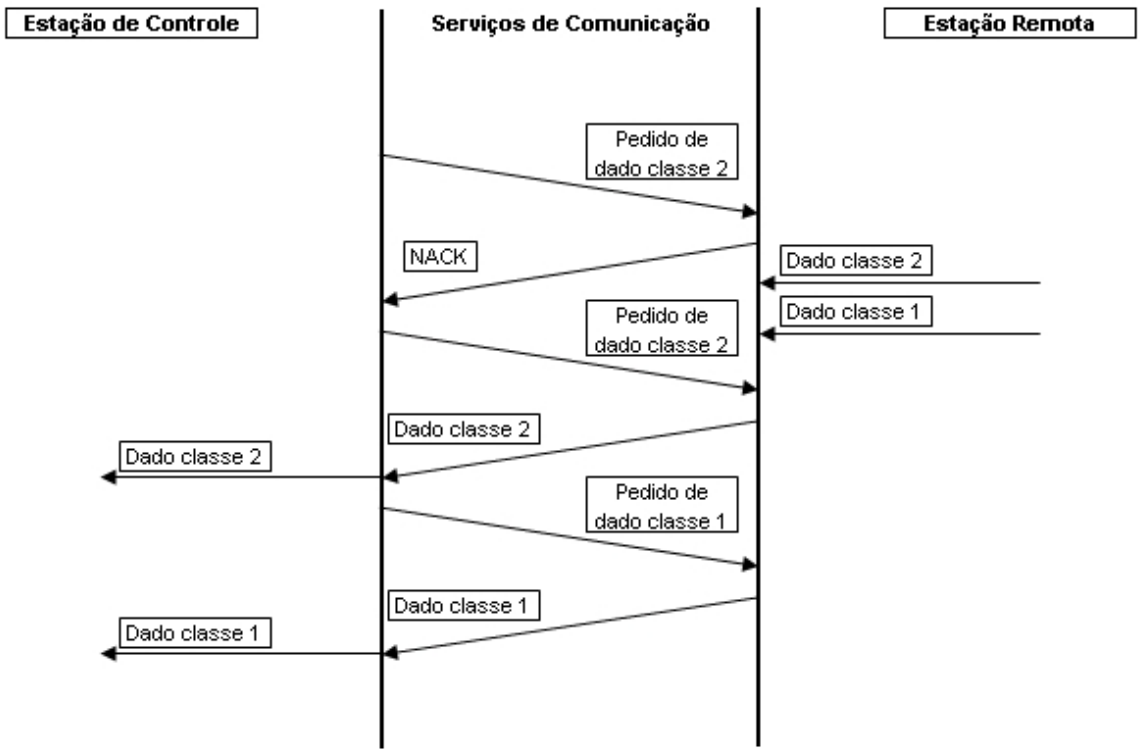
A aquisição de dados por "polling" é usada em sistemas SCADA operando com procedimentos de transmissão não-balanceados com o objetivo de atualizar a estação de controle com o atual estado das variáveis de processo nas estações remotas. A estação de controle realiza o "polling" interrogando as remotas seqüencialmente, e estas por sua vez somente podem transmitir quando for solicitado pela estação de controle. Normalmente, cada mensagem de "polling" é transmitida em intervalos fixos de tempo, podendo sua duração variar de um sistema para o outro dependendo da implementação utilizada. Procura-se utilizar um tempo não muito curto, visando o não congestionamento do meio de transmissão com uma quantidade excessiva de dados, e também não é interessante que o tempo seja muito longo para não haja um acúmulo muito grande de dados esperando para serem transmitidos pela remota.

O primeiro procedimento apresentado na figura abaixo mostra um pedido de dados de classe 1, que é uma solicitação feita à remota por dados desta classe. Este pedido é respondido com um NACK, indicando a não existência de informações de classe 1 a serem transmitidas pela remota. O próximo pedido enviado é um pedido de dados de classe 2, que assim como o anterior é uma solicitação de dados feita à remota, porém são solicitados dados de classe 2.

Como podemos verificar entre os dois pedidos feitos pela estação de controle, foram gerados na remota dados de ambas as classes. Como a estação de controle está solicitando dados de

classe 2, a remota irá responder com os dados disponíveis desta classe e irá indicar, através do bit ACD no byte de controle, a presença de dados de classe 1 disponíveis. A estação de controle irá então no seu próximo "polling" pedir por dados desta classe através de um pedido de dados de classe 1.

Figura II.26 – Aquisição de dados por "polling"



Como já foi citado anteriormente, após os procedimentos de inicialização da estação de controle ou da remota, normalmente seguem-se dois tipos de mensagem. O primeiro tipo é a mensagem de "general interrogation" que será descrita a seguir, o outro tipo será descrito posteriormente.

II.3.5.6 – "General Interrogation"

O processo de interrogação da remota é usado para atualizar a estação de controle após uma inicialização interna ou quando foi detectado algum tipo de erro ou perda de informação. A função de "general interrogation" da estação de controle solicita junto às remotas a transmissão dos valores atuais de **todas** as variáveis de processo.

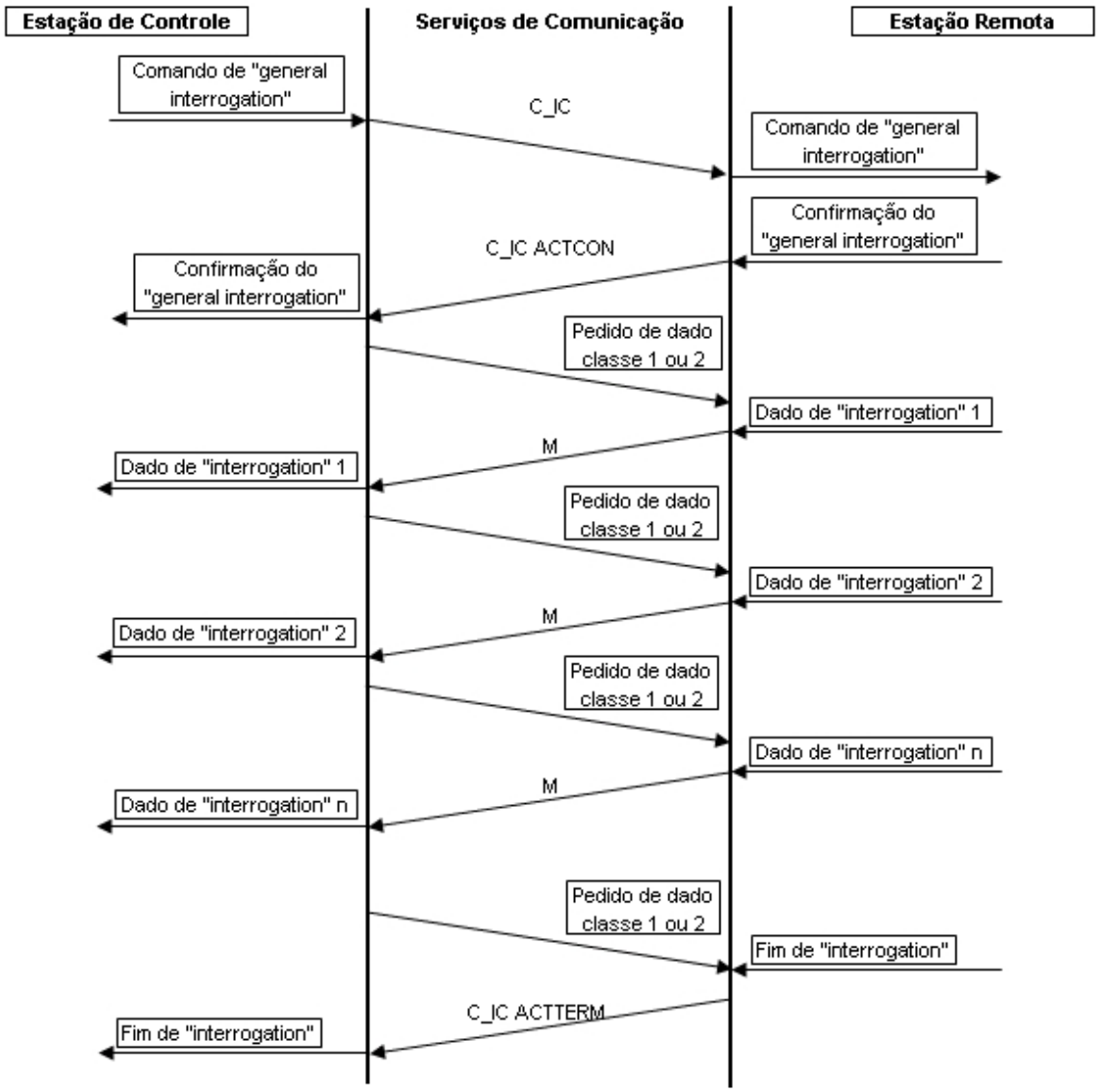
Após o recebimento de um "general interrogation", a remota transmite a informação solicitada. A quantidade de dados que serão enviadas normalmente é conhecida pelas funções de aplicação das duas estações. Neste caso, a quantidade de dados transmitida pode ser checada pela estação de controle, e o término da transmissão dos dados é determinado pelo recebimento de todas as informações esperadas. Caso a quantidade de dados a serem transmitidos não seja conhecida pela estação de controle, a remota pode indicar o fim do envio os dados de "general interrogation" através de uma mensagem de fim de "general interrogation".

O procedimento de "general interrogation" mostrado na figura abaixo pode ser descrito da seguinte forma. Primeiro, a estação de controle envia uma mensagem de comando "general interrogation" para a remota. Esta, por sua vez, retorna uma mensagem de confirmação do comando enviado anteriormente. A partir deste momento, a remota irá enviar, seqüencialmente, todos os dados referentes às variáveis de estados para a estação de controle. Nos sistemas não-balanceados, a estação de controle efetua diversos "pollings", sendo que cada um destes é respondido com informações das variáveis de estado da remota. Caso a estação de controle não tenha o conhecimento do total de dados a ser coletado neste procedimento de "general interrogation", a última mensagem recebida indicará o fim da

transmissão de todos os dados presentes na remota. A partir desse ponto, o processo que aquisição de dados continua a funcionar com descrito anteriormente.

Iremos verificar no capítulo seguinte, que as mensagens de resposta ao pedido de "general interrogation" são diferenciadas das mensagens de resposta ao pedidos de "polling". Um campo presente nas mensagens indica qual é a causa da transmissão dos dados solicitados. Para o primeiro caso, a causa da transmissão será o "polling", enquanto que para o segundo a causa será "general interrogation".

Figura II.27 – Procedimento de "general interrogation"



Iremos descrever agora o segundo tipo de mensagem que normalmente é transmitida após uma inicialização da estação de controle ou da estação remota.

II.3.5.7 – Sincronismo do Clock

O relógio da remota precisa estar sincronizado com o relógio da estação de controle. Isto fará com que a transmissão de mensagens com "time tag" ou tag de tempo possa ser válida e apresente informações precisas e úteis. O relógio é inicialmente sincronizado pela estação de controle após a inicialização do sistema, e sincronizado novamente periodicamente através de mensagens de "clock synchronization" (C_CS ACT).

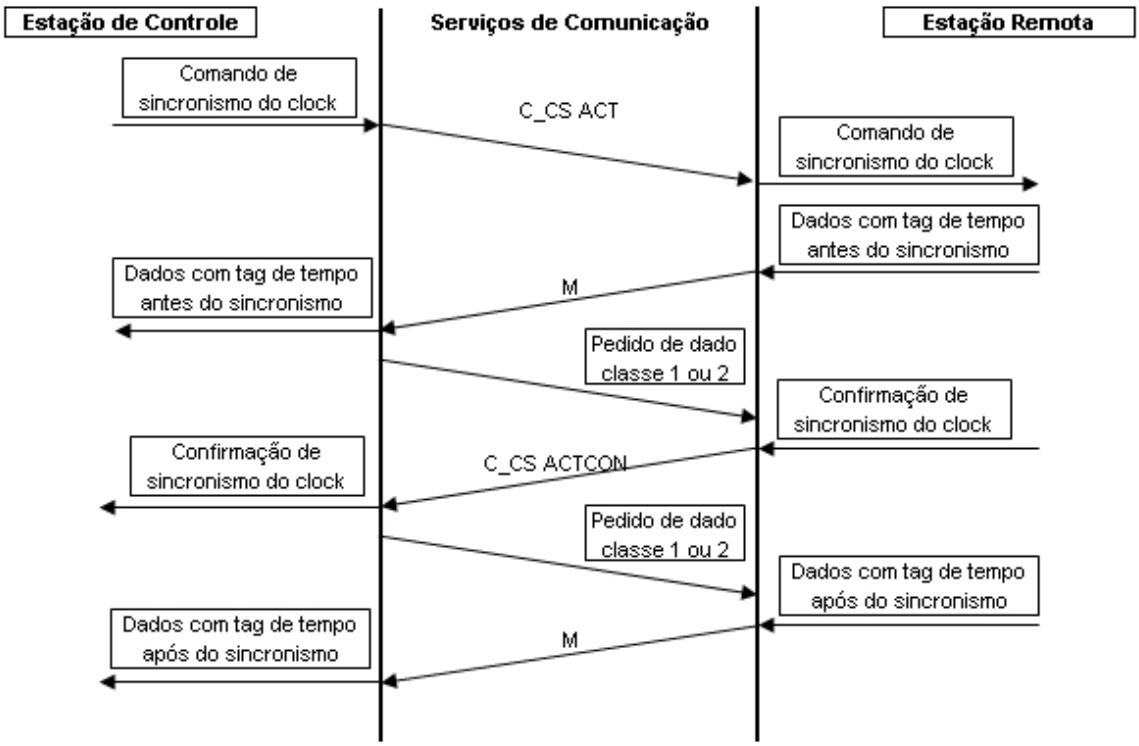
As mensagens de C_CS ACT contêm informações da hora atual do relógio, ou seja, informação de data e hora com a necessária resolução de tempo no momento em que o primeiro bit de C_CS ACT é transmitido. A informação de tempo enviada precisa ser corrigida pela remota. O valor da correção é determinado pelo tempo de transmissão somado ao produto entre o tamanho da mensagem de C_CS ACT e a taxa de transmissão da mesma. A execução desta operação nas remotas depende de requisitos particulares dos processos e não está sujeita a padronização. Após a execução deste comando, a remota retorna uma mensagem de C_CS ACTCON. Esta contém a informação do tempo local antes do sincronismo menos o valor da correção de tempo. Esta mensagem é transmitida após as mensagens armazenas com tag de tempo que esperam para serem transmitidas. Os eventos que venham a ocorrer após o sincronismo do relógio são transmitidos após a mensagem de C_CS ACTCON.

As remotas esperam o recebimento dos comandos de sincronismo dentro de intervalos específicos de tempo. Quando este comando não é recebido dentro deste intervalo de tempo,

que é conseqüência da precisão do relógio local e desvios de tempo tolerados, a remota caracteriza todos os objetos de informação com tag de tempo como suspeitos. Esta marca também é inserida após uma inicialização local da remota, antes que haja o recebimento de um comando de sincronismo válido pela estação de controle. Dados com tag de tempo gerados após o recebimento de um comando válido de sincronismo são transmitidos sem a marca.

O procedimento descrito pode ser ilustrado na figura abaixo.

Figura II.28 – Procedimento de sincronismo do clock



II.3.5.8 – Transmissão de comandos

Um comando é usado em sistemas de controle e telemetria para provocar mudanças de estado em um equipamento, ou seja, os comandos são utilizados para guiar um processo para uma determinada direção.

Os comandos podem ser iniciados por um operador ou por procedimentos automáticos de supervisão nas estações de controle. Alguns componentes ou equipamentos que podem ser controlados são:

- chaves ou contatos elétricos;
- início ou fim de um processo de controle;
- execução de alguma manobra em uma seqüência de controle;
- marcações, limites de alarmes, parâmetros específicos, etc.

Existem dois tipos de procedimentos para a transmissão de comandos, a saber:

1. Comandos diretos e
2. Comando de seleção e execução

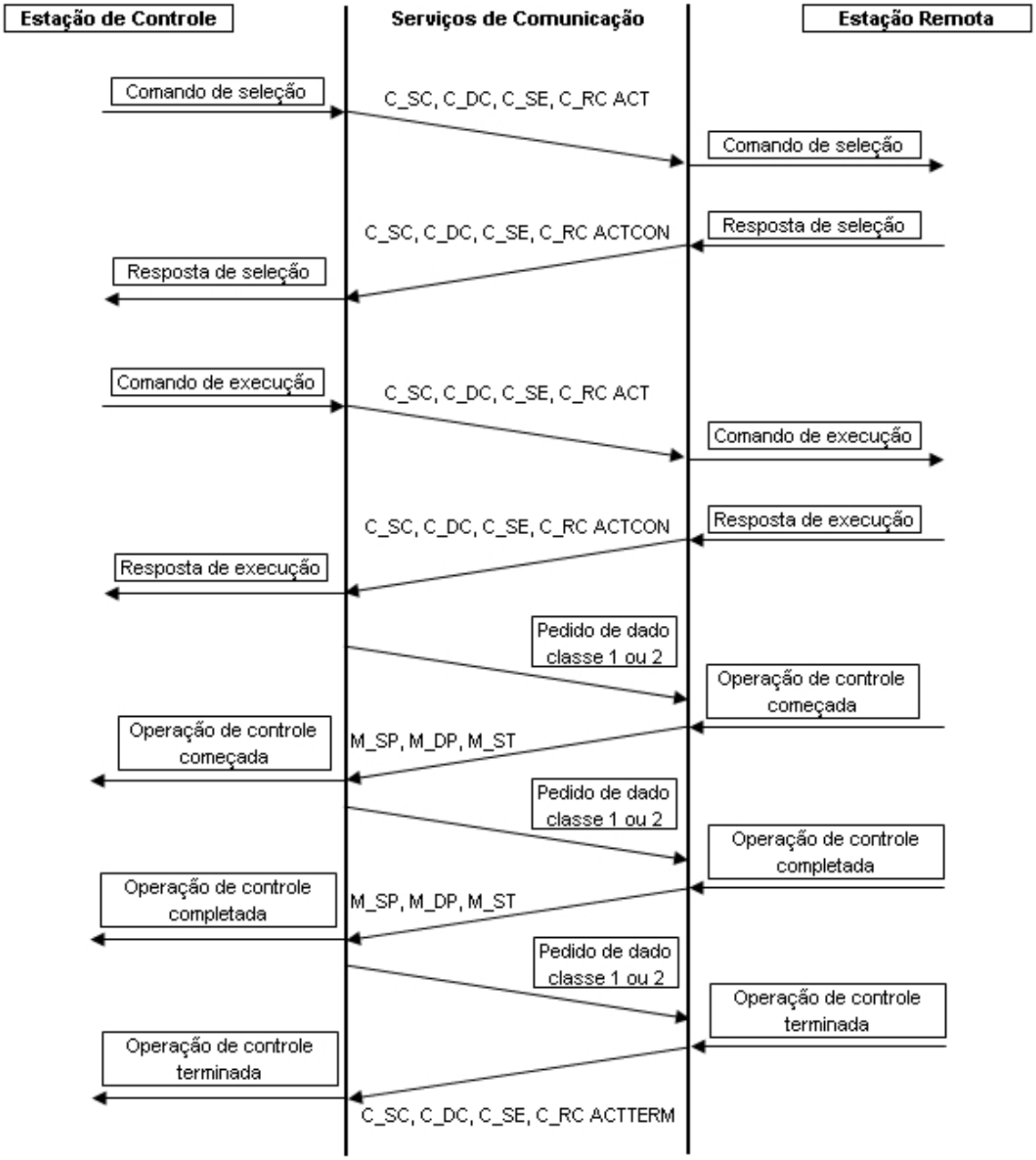
Comandos diretos são usados em estações de controle para a execução imediata de tarefas na remota. As funções de aplicação da remota verificam, por questões de segurança, a validade dos comandos recebidos e executa a operação.

O procedimento de seleção e execução é usado pelas estações de controle para preparar a remota para a execução de um comando, verificando se a operação correta foi preparada e só

então executar o comando. A verificação pode ser executada por um operador ou pela aplicação. A remota não dará início à execução do comando enquanto não receber a mensagem de execução.

O exemplo mostrado na figura abaixo mostra um procedimento de envio de mensagens de comando da estação de controle para a remota. Primeiramente, a estação de controle envia uma mensagem de comando de seleção. Caso a estação remota esteja pronta para receber o comando esta retornará uma mensagem de confirmação de seleção. O processo de seleção pode ser cancelado pela estação de controle com o envio de um comando de break. Caso a estação de controle tenha interesse na execução do comando indicado esta enviará uma mensagem de execução de comando que por sua vez será confirmada pela remota. Opcionalmente, a remota poderá enviar mensagens informando que a operação solicitada foi começado, informando o estado atual das variáveis do processo, que a operação foi completada, informando o estado final das variáveis de estado e também que a execução do comando foi concluída.

Figura II.29 – Envio de mensagens de comando



No próximo capítulo, iremos definir mais detalhadamente os campos que compõem as mensagens citadas anteriormente identificando suas variações e peculiaridades.

II.3.6. IEC 870-5-101

Esta norma define um padrão para sistemas de controle e telemetria que permite a interoperabilidade entre equipamentos de controle. Os padrões definidos nesta norma são compatíveis com os padrões definidos nas normas anteriores. As especificações aqui descritas buscam apresentar um perfil funcional para a realização de operações básicas de telemetria. A seguir, serão feitas considerações específicas sobre as informações apresentadas nas normas precedentes, ou seja, serão definidos os métodos de transmissão a serem utilizados, as mensagens e os campos que as compõem, assim como outras características necessárias para a implementação do protocolo, características essas que não tinham sido definidas com mais detalhes nas normas anteriores pois não se destinavam a este fim.

II.3.6.1 – Camada de link

A norma IEC 870-5-2 oferece uma variedade de procedimentos de transmissão que usam o byte de controle e o byte de endereço. Os links entre as estações podem operar no modo balanceado ou não balanceado. As funções apropriadas para o byte de controle são especificadas em ambos os modos de operação.

Caso os links entre uma central de controle (estação de controle) e uma ou mais remotas dividam o mesmo canal físico de transmissão, então estes links necessitam operar no modo não-balanceado, para impedir a possibilidade de duas estações tentarem transmitir ao mesmo tempo. A seqüência na qual as várias estações remotas recebem acesso de transmissão no canal é então determinado pela estação de controle, pois como vimos anteriormente, no modo não-balanceado a remota somente pode transmitir caso solicitado pela estação de controle.

A norma especifica quando deve ser utilizado um método de transmissão balanceado ou não-balanceado, junto com as devidas funções de link a serem utilizadas. Especifica também um endereço não ambíguo para cada link. Cada endereço pode ser único dentro de um determinado sistema, ou dentro de um grupo de links que compartilham de um canal em comum.

A norma pode definir um formato de frame, escolhido dentre os apresentados na IEC 870-5-1. O formato escolhido deve fornecer a integridade necessária dos dados, aliado com a máxima eficiência disponível para um nível aceitável de conveniência de implementação. Esta norma admite exclusivamente o formato de frame FT 1.2. Formatos com tamanho fixo e variável são admitidos, assim como o caractere único de controle.

O tamanho máximo dos frames da camada de link é definido como um parâmetro fixo do sistema. Caso necessário, o tamanho máximo para cada direção pode ser diferente. O frame de tamanho fixo não possui o campo "link user data", logo seu tamanho fica fixado com sendo igual a 5 bytes.

Nos sistemas de transmissão não-balanceados, as remotas são sempre secundárias (escravas). O centro de controle é primário (mestre).

O bit RES presente no byte de controle não é utilizado.

O campo de endereço pode possuir 1 ou 2 bytes, determinado por um parâmetro fixo do sistema. O número de endereço para broadcast é 255 para o caso de um campo de 1 byte ou 65535 para o caso de 2 bytes. Não são definidos endereços de grupos.

Nos sistemas de "polling", o procedimento básico de transmissão utiliza o serviço de REQUEST/RESPOND de função de código igual a 11 (pedido de dados de classe 2), de acordo com a **tabela II.1**. Transmissões de classe 1 são indicadas através o bit ACD, como definido na IEC 870-5-2. Remotas que não possuírem dados de classe 2 precisam responder com função de código igual a 9, indicando que o dado solicitado não está disponível.

II.3.6.2 – Camada de aplicação

A norma deve definir ASDUs apropriadas de uma estrutura geral apresentada na IEC 870-5-3. Essas ASDUs são construídas usando as especificações e definições apresentadas na IEC 870-5-4. A IEC 870-5-3 descreve as unidades de dados de nível de aplicação no frames de transmissão. A unidade de dados do protocolo da aplicação (LPDU) contido nesta norma contém não mais de uma ASDU.

A ASDU é composta por um identificador de unidade de dados (DUI) e um ou mais objetos de informação (IO).

O identificador de unidade de dados possui sempre a mesma estrutura para todas as ASDUs. Os objetos de informação são sempre da mesma estrutura e tipo de acordo com o que está definido no campo identificador de tipo (TI).

A estrutura de um identificador de unidade de dados é:

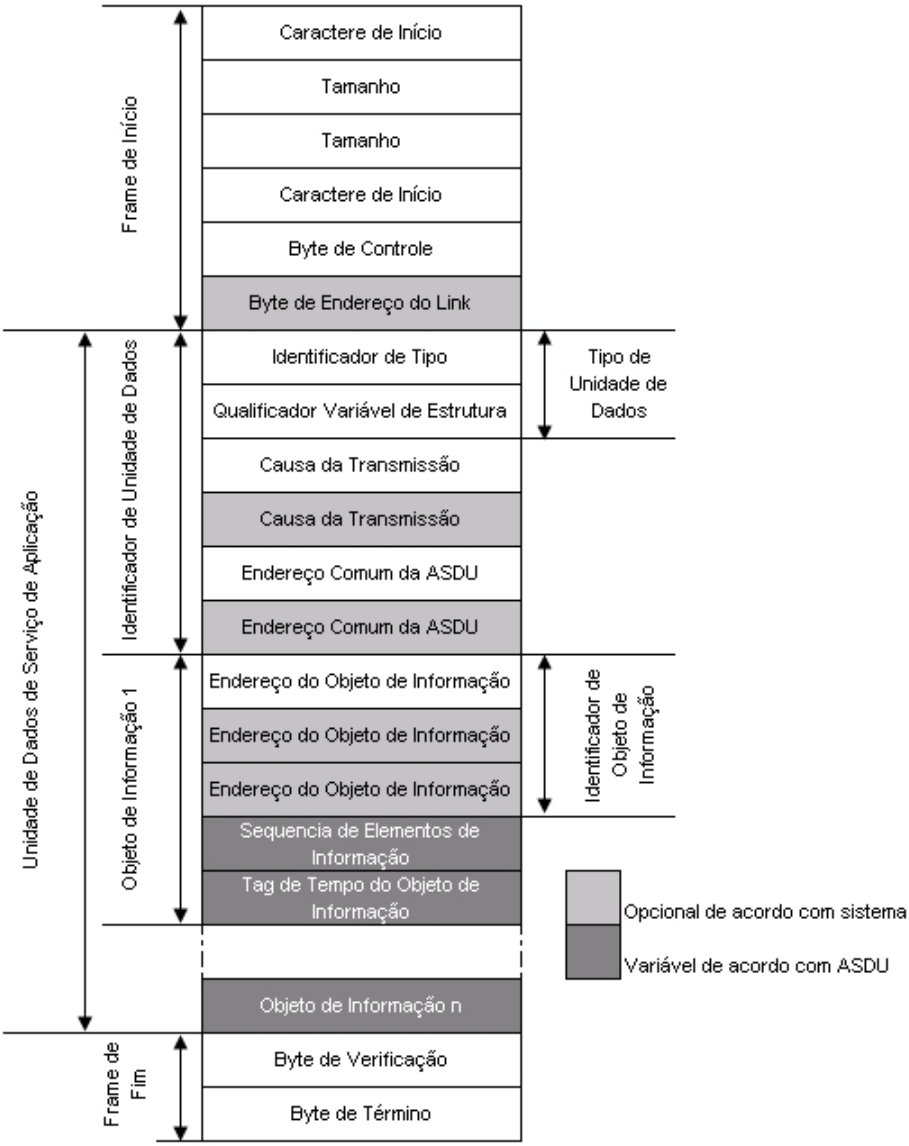
- um byte – identificador de tipo (TI);
- um byte – qualificador variável de estrutura (VSQ);
- um ou dois bytes – causa da transmissão (COT);
- um ou dois bytes – endereço comum da ASDU (CAA).

O tamanho do endereço comum da ASDU é determinado por um parâmetro fixo do sistema que pode ser igual a 1 ou 2 bytes. O endereço comum é o endereço da estação, que pode ser estruturado, para permitir o endereçamento de toda a estação, ou somente um setor em particular. Não está presente o campo de tamanho ASDU. Como cada frame somente possui uma ASDU seu tamanho pode ser determinado pelo tamanho do frame.

Tags de tempo, caso presentes, pertencem sempre a um único objeto de informação. Um objeto de informação é formado por um identificador de objeto de informação, uma seqüência de elementos de informação e, caso presente, um tag de tempo do objeto de informação.

O identificador de objeto de informação consiste apenas do endereço do objeto de informação. Na maioria dos casos o endereço comum da ASDU aliado ao endereço do objeto de informação distingue o conjunto de seqüências de elementos de informação dentro de um sistema específico. A combinação dos dois endereços precisa ser única por sistema. O identificador de tipo não faz parte do endereço comum nem do endereço do objeto de informação.

Figura II.30 – Estrutura da mensagem de tamanho variável

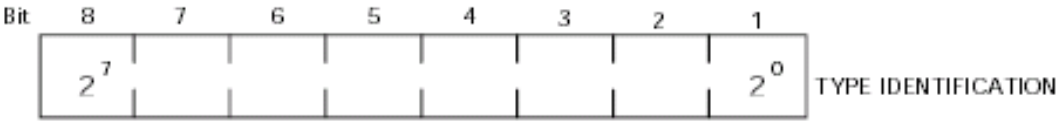


Os tamanhos e os conteúdos dos campos de informação mostrados na ASDU da figura acima serão descritos em seguida de acordo com as regras contidas na IEC 870-5-4.

II.3.6.2.1 – Identificador de tipo

O primeiro byte mostrado na ASDU anterior define a estrutura, o tipo e o formato dos objetos de informação que serão mostrados posteriormente.

Figura II.31 – Identificador de Tipo



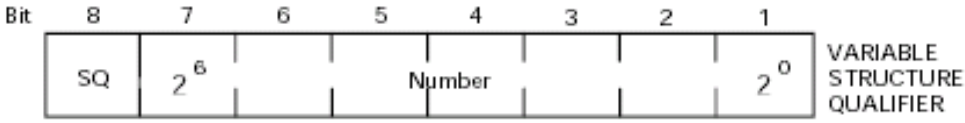
Os objetos de informação com ou sem tags de tempo são distinguidos por diferentes números de identificador de tipo. O valor 0 não é usado. O range de valores assumidos por este campo por ser descrito da seguinte forma:

Identificador de Tipo	UI8[1..8]<1..255>
<1..127>	para definições contidas nesta norma
<128..135>	reservado para mensagens de roteamento
<136..255>	para uso especial

II.3.6.2.2 – Qualificador variável de estrutura

O segundo byte do identificador de unidade de dados da ASDU define o qualificador variável de estrutura.

Figura II.32 – Qualificador variável de estrutura

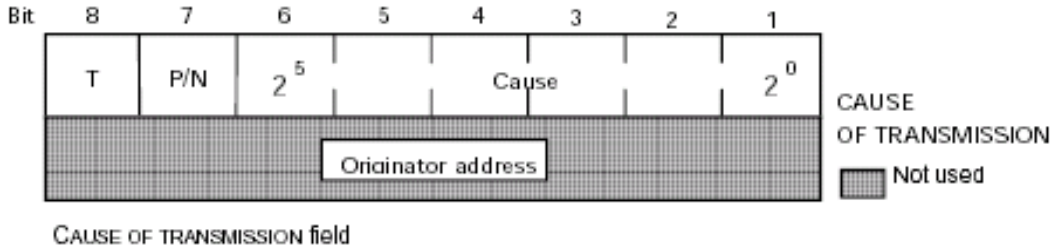


Qualificador Variável de Estrutura	CP8{número,SQ}
número=N	UI7[1..7] <0..127>
<0>	ASDU não contém Objetos de Informação
<1..127>	número de Objetos de Informação ou Elementos
SQ=único/sequencia	BS1[8] <0..1>
<0>	endereçamento de um elemento individual ou combinação de elementos em um número de Objetos de Informação do mesmo tipo
<1>	endereçamento de uma Seqüência de Elementos de Informação em um objeto
SQ<0> e N<0..127>	número de Objetos de Informação
SQ<1> e N<0..127>	número de Elementos de Informação de um único objeto por ASDU

II.3.6.2.3 – Causa da transmissão

O terceiro byte do identificador de unidade de dados da ASDU define o campo causa da transmissão, responsável por informar a causa da transmissão da mensagem em questão.

Figura II.33 – Causa da transmissão



Causa da Transmissão	CP16{Causa,P/N,T,Endereço Originação}
Causa	UI6[1..6] <0..63>
<0>	não definida
<1..63>	número da causa
<1..47>	para definições contidas nesta norma
<48..63>	para uso especial
P/N	BS1[8] <0..1>
<0>	não teste
<1>	teste
Endereço Originação (opcional)	UI8[9..16]
<0>	padrão
<1..255>	número do endereço de originação

Este campo direciona a ASDU para uma aplicação específica para ser processada. O bit P/N indica a confirmação positiva ou negativa da ativação requisitada pela função de aplicação primária. Caso seja irrelevante, este bit assume o valor 0. O bit T é utilizado para indicar que a mensagem transmitida representa um teste. Isto ocorre quando se deseja testar a transmissão e os equipamentos sem provocar a execução de processos que poderiam ser executados devido às mensagens enviadas.

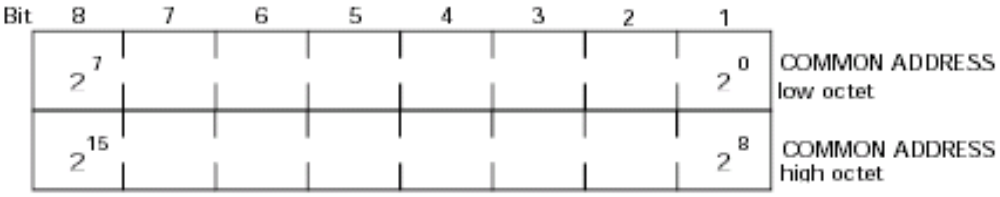
Algumas mensagens enviadas pela estação de controle podem receber respostas que são um espelho das mensagens enviadas. A diferenciação entre essas duas mensagens pode ser feita

pelas diferentes causas de transmissão. Por exemplo, uma mensagem enviada pela estação de controle, com causa de transmissão igual a “activation”, pode receber como resposta uma mensagem idêntica, onde o campo de causa de transmissão da mensagem recebida é igual a “activation confirmation”.

II.3.6.2.4 – Endereço comum da ASDU

O quarto byte e opcionalmente o quinto do identificador de unidade de dados da ASDU define o endereço da estação. O tamanho deste campo (um ou dois bytes) é um parâmetro fixo de acordo com o sistema e deve ser acordado entre as partes.

Figura II.34 – Endereço comum da ASDU



Para Endereço Comum da ASDU de 1 byte

Endereço Comum da ASDU	UI8[1..8] <0..255>
<0>	não usado
<1..254>	endereço da estação
<255>	endereço global

Para Endereço Comum da ASDU de 2 bytes

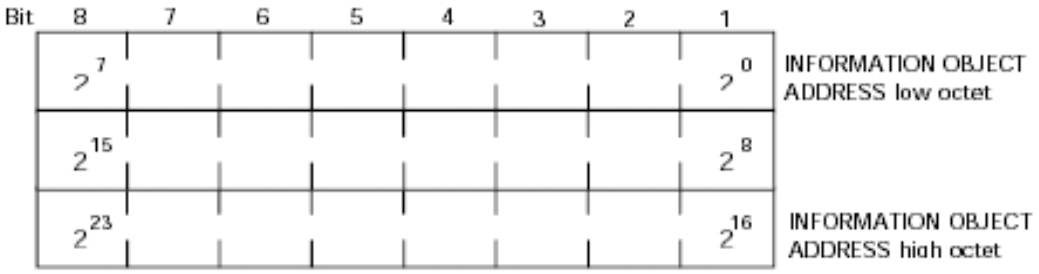
Endereço Comum da ASDU	UI16[1..16] <0..65535>
<0>	não usado
<1..65534>	endereço da estação
<65535>	endereço global

Este campo está associado com todos os objetos em uma ASDU. O endereço global é um endereço de broadcast direcionado a todas as estações em um sistema específico.

II.3.6.2.5 – Endereço do objeto de informação

O primeiro, e opcionalmente o segundo e o terceiro bytes do objeto de informação, são utilizados para definir os endereços dos componentes do sistema das estações remotas como, por exemplo, o endereço associado a uma chave digital ou a um valor de tensão em uma linha de transmissão. O tamanho deste campo (um, dois ou três bytes) é um parâmetro fixo de acordo com o sistema e deve ser acordado entre as partes envolvidas. Este campo é utilizado com endereço de destino em mensagens enviadas da estação de controle para a remota, ou como um endereço de origem no sentido contrário.

Figura II.35 – Endereço do objeto de informação



Para Endereço do Objeto de Informação de 1 byte

Endereço do Objeto de Informação	UI8[1..8] <0..255>
<0>	endereço irrelevante
<1..255>	endereço do elemento de informação

Para Endereço do Objeto de Informação de 2 bytes

Endereço do Objeto de Informação	UI16[1..16] <0..65535>
<0>	endereço irrelevante
<1..65535>	endereço do elemento de informação

Para Endereço do Objeto de Informação de 3 bytes

Endereço do Objeto de Informação	UI24[1..24] <0..16777215>
<0>	endereço irrelevante
<1..16777215>	endereço do elemento de informação

O terceiro byte somente é utilizado no caso de um endereço do objeto de informação estruturado, de forma a definir endereços únicos dentro de um sistema específico. Caso este campo não seja relevante ele assume o valor 0.

A seguir, serão especificadas as mensagens definidas na norma. Normas futuras poderão definir outros tipos de mensagens com diferentes números para o identificador de tipo. Vale ressaltar, que somente algumas mensagens definidas nesta norma serão descritas a seguir. Serão detalhadas as mensagens que melhor se encaixam na implementação do sistema desenvolvido, visando ilustrar o funcionamento de um sistema de supervisão e controle na área elétrica. Serão mostradas algumas mensagens de transmissão de dados digitais, medidas e mensagens de comando.

II.3.6.3 – Unidades de Dados de Serviço de Aplicação (ASDUs)

A seguir, serão ilustrados alguns tipos de mensagens comuns que são utilizadas nas trocas de mensagens em processos de telemetria e supervisão e controle. Estas representam apenas uma parcela das mensagens definidas na norma, e serão apresentadas para exemplificar os diferentes tipos de dados que estas podem conter de acordo com o tipo da mensagem.

II.3.6.3.1 – Informação de estado sem tag de tempo (M_SP_NA_1), SQ = 0

Figura II.36 – Mensagem M_SP_NA_1 com SQ = 0

0	0	0	0	0	0	0	1	Identificador de Tipo
0	Número de objetos							Qualificador Variável da Estrutura
Causa da Transmissão								
Endereço Comum da ASDU								
Endereço do Objeto de Informação								Objeto de Informação 1
IV	NT	SB	BL	0	0	0	SPI	SIQ
⋮								
Endereço do Objeto de Informação								Objeto de Informação N
IV	NT	SB	BL	0	0	0	SPI	SIQ

SIQ	8 bits
SPI	1 bit
<0>	OFF
<1>	ON
RES	3 bits - Reservado
BL	1 bit
<0>	não bloqueado
<1>	bloqueado
SB	1 bit
<0>	não substituído
<1>	substituído
NT	1 bit
<0>	atualizado
<1>	não atualizado
IV	1 bit
<0>	válido
<1>	inválido

II.3.6.3.2 – Informação de estado sem tag de tempo (M_SP_NA_1), SQ = 1

Figura II.37 – Mensagem M_SP_NA_1 com SQ = 1

0	0	0	0	0	0	0	1	Identificador de Tipo
1	Número de objetos							Qualificador Variável da Estrutura
Causa da Transmissão								
Endereço Comum da ASDU								
Endereço do Objeto de Informação								Objeto de Informação
IV	NT	SB	BL	0	0	0	SPI	SIQ 1
⋮								
IV	NT	SB	BL	0	0	0	SPI	SIQ n

SIQ	8 bits
SPI	1 bit
<0>	OFF
<1>	ON
RES	3 bits - Reservado
BL	1 bit
<0>	não bloqueado
<1>	bloqueado
SB	1 bit
<0>	não substituído
<1>	substituído
NT	1 bit
<0>	atualizado
<1>	não atualizado
IV	1 bit
<0>	válido
<1>	inválido

II.3.6.3.3 – Informação de estado com tag de tempo (M_SP_TA_1)

Figura II.38 – Mensagem M_SP_TA_1

0	0	0	0	0	0	1	0	Identificador de Tipo
0	Número de objetos							Qualificador Variável da Estrutura
Causa da Transmissão								
Endereço Comum da ASDU								
Endereço do Objeto de Informação								Objeto de Informação 1
IV	NT	SB	BL	0	0	0	SPI	SIQ
CP24Time2a								Tag de tempo 3 bytes
⋮								⋮
Endereço do Objeto de Informação								Objeto de Informação N
IV	NT	SB	BL	0	0	0	SPI	SIQ
CP24Time2a								Tag de tempo 3 bytes

SIQ	8 bits
SPI	1 bit
<0>	OFF
<1>	ON
RES	3 bits - Reservado
BL	1 bit
<0>	não bloqueado
<1>	bloqueado
SB	1 bit
<0>	não substituído
<1>	substituído
NT	1 bit
<0>	atualizado
<1>	não atualizado
IV	1 bit
<0>	válido
<1>	inválido
Tag de Tempo	Definido na norma IEC 870-5-4

II.3.6.3.4 – Informação de estado duplo sem tag de tempo (M_DP_NA_1), SQ = 0

Figura II.39 – Mensagem M_DP_NA_1 com SQ = 0

0	0	0	0	0	0	1	1	Identificador de Tipo
0	Número de objetos							Qualificador Variável da Estrutura
Causa da Transmissão								
Endereço Comum da ASDU								
Endereço do Objeto de Informação								Objeto de Informação 1
IV	NT	SB	BL	0	0	DPI		DIQ
⋮								
Endereço do Objeto de Informação								Objeto de Informação N
IV	NT	SB	BL	0	0	DPI		DIQ

DIQ	8 bits
DPI	2 bit
<0>	estado não determinado ou intermediário
<1>	OFF
<2>	ON
<3>	estado indeterminado
RES	2 bits - Reservado
BL	1 bit
<0>	não bloqueado
<1>	bloqueado
SB	1 bit
<0>	não substituído
<1>	substituído
NT	1 bit
<0>	atualizado
<1>	não atualizado
IV	1 bit
<0>	válido
<1>	inválido

II.3.6.3.5 – Informação de estado duplo sem tag de tempo (M_DP_NA_1), SQ = 1

Figura II.40 – Mensagem M_DP_NA_1 com SQ = 1

0	0	0	0	0	0	1	1	Identificador de Tipo
1	Número de objetos							Qualificador Variável da Estrutura
Causa da Transmissão								
Endereço Comum da ASDU								
Endereço do Objeto de Informação								Objeto de Informação
IV	NT	SB	BL	0	0	DPI		DIQ 1
⋮								
IV	NT	SB	BL	0	0	DPI		DIQ n

DIQ	8 bits
DPI	2 bit
<0>	estado não determinado ou intermediário
<1>	OFF
<2>	ON
<3>	estado indeterminado
RES	2 bits - Reservado
BL	1 bit
<0>	não bloqueado
<1>	bloqueado
SB	1 bit
<0>	não substituído
<1>	substituído
NT	1 bit
<0>	atualizado
<1>	não atualizado
IV	1 bit
<0>	válido
<1>	inválido

II.3.6.3.6 – Informação de estado duplo com tag de tempo (M_DP_TA_1)

Figura II.41 – Mensagem M_DP_TA_1

0	0	0	0	0	1	0	0	Identificador de Tipo
0	Número de objetos							Qualificador Variável da Estrutura
Causa da Transmissão								
Endereço Comum da ASDU								
Endereço do Objeto de Informação								Objeto de Informação 1
IV	NT	SB	BL	0	0	DPI		DIQ
CP24Time2a								Tag de tempo 3 bytes
⋮								⋮
Endereço do Objeto de Informação								Objeto de Informação N
IV	NT	SB	BL	0	0	DPI		DIQ
CP24Time2a								Tag de tempo 3 bytes

DIQ	8 bits
DPI	2 bit
<0>	estado não determinado ou intermediário
<1>	OFF
<2>	ON
<3>	estado indeterminado
RES	2 bits - Reservado
BL	1 bit
<0>	não bloqueado
<1>	bloqueado
SB	1 bit
<0>	não substituído
<1>	substituído
NT	1 bit
<0>	atualizado
<1>	não atualizado
IV	1 bit
<0>	válido
<1>	inválido
Tag de Tempo	Definido na norma IEC 870-5-4

II.3.6.3.7 – Informação de medida normalizada (M_ME_NA_1), SQ=0

Figura II.42 – Mensagem M_ME_NA_1 com SQ = 0

0	0	0	0	1	0	0	1		Identificador de Tipo
0	Número de objetos								Qualificador Variável da Estrutura
Causa da Transmissão									
Endereço Comum da ASDU									
Endereço do Objeto de Informação									
Valor								NVA	Objeto de Informação 1
S	Valor								
IV	NT	SB	BL	0	0	0	OV	QDS	
⋮									
Endereço do Objeto de Informação									
Valor								NVA	Objeto de Informação N
S	Valor								
IV	NT	SB	BL	0	0	0	OV	QDS	

QDS	8 bits
OV	1 bit
<0>	sem overflow
<1>	overflow
RES	3 bits - Reservado
BL	1 bit
<0>	não bloqueado
<1>	bloqueado
SB	1 bit
<0>	não substituído
<1>	substituído
NT	1 bit
<0>	atualizado
<1>	não atualizado
IV	1 bit
<0>	válido
<1>	inválido
NVA	Definido na norma IEC 870-5-4

II.3.6.3.8 – Informação de medida normalizada (M_ME_NA_1), SQ=1

Figura II.43 – Mensagem M_ME_NA_1 com SQ = 1

0	0	0	0	1	0	0	1	Identificador de Tipo	
1	Número de objetos							Qualificador Variável da Estrutura	
Causa da Transmissão									
Endereço Comum da ASDU									
Endereço do Objeto de Informação									
Valor								NVA 1	Objeto de Informação
S	Valor								
IV	NT	SB	BL	0	0	0	OV	QDS 1	
⋮									
Valor								NVA n	
S	Valor								
IV	NT	SB	BL	0	0	0	OV	QDS n	

QDS	8 bits
OV	1 bit
<0>	sem overflow
<1>	overflow
RES	3 bits - Reservado
BL	1 bit
<0>	não bloqueado
<1>	bloqueado
SB	1 bit
<0>	não substituído
<1>	substituído
NT	1 bit
<0>	atualizado
<1>	não atualizado
IV	1 bit
<0>	válido
<1>	inválido
NVA	Definido na norma IEC 870-5-4

II.3.6.3.9 – Informação de medida normalizada com tag de tempo (M_ME_TA_1)

Figura II.44 – Mensagem M_ME_TA_1

0	0	0	0	1	0	1	0	Identificador de Tipo	
0	Número de objetos							Qualificador Variável da Estrutura	
Causa da Transmissão									
Endereço Comum da ASDU									
Endereço do Objeto de Informação									
Valor								NVA	Objeto de Informação 1
S	Valor								
IV	NT	SB	BL	0	0	0	OV	QDS	
CP24Time2a								Tag de tempo 3 bytes	
⋮								⋮	
Endereço do Objeto de Informação									
Valor								NVA	Objeto de Informação N
S	Valor								
IV	NT	SB	BL	0	0	0	OV	QDS	
CP24Time2a								Tag de tempo 3 bytes	

QDS	8 bits
OV	1 bit
<0>	sem overflow
<1>	overflow
RES	3 bits - Reservado
BL	1 bit
<0>	não bloqueado
<1>	bloqueado
SB	1 bit
<0>	não substituído
<1>	substituído
NT	1 bit
<0>	atualizado
<1>	não atualizado
IV	1 bit
<0>	válido
<1>	inválido
NVA	Definido na norma IEC 870-5-4
Tag de Tempo	Definido na norma IEC 870-5-4

II.3.6.3.10 – Comando simples (C_SC_NA_1)

Figura II.45 – Mensagem C_SC_NA_1

0	0	1	0	1	1	0	1	Identificador de Tipo
0	0	0	0	0	0	0	1	Qualificador Variável da Estrutura
Causa da Transmissão								
Endereço Comum da ASDU								
Endereço do Objeto de Informação								Objeto de Informação
S/E	QU					0	SCS	SCO

SCO	8 bits
SCS	1 bit
<0>	OFF
<1>	ON
RES	1 bit - Reservado
QU	5 bits
<0>	sem informação adicional
<1>	pulso de duração curta
<2>	pulso de duração longa
<3>	saída constante
<4..8>	reservado para futuras definições desta norma
<9..15>	reservado
<16..31>	reservado
SE	1 bit
<0>	executar
<1>	selecionar

II.3.6.3.11 – Comando duplo (C_DC_NA_1)

Figura II.46 – Mensagem C_DC_NA_1

0	0	1	0	1	1	1	0	Identificador de Tipo
0	0	0	0	0	0	0	1	Qualificador Variável da Estrutura
Causa da Transmissão								
Endereço Comum da ASDU								
Endereço do Objeto de Informação								Objeto de Informação
S/E	QU					DCS		DCO

DCO	8 bits
DCS	2 bits
<0>	não permitido
<1>	OFF
<2>	ON
<3>	não permitido
QU	5 bits
<0>	sem informação adicional
<1>	pulso de duração curta
<2>	pulso de duração longa
<3>	saída constante
<4..8>	reservado para futuras definições desta norma
<9..15>	reservado
<16..31>	reservado
SE	1 bit
<0>	executar
<1>	selecionar

II.3.6.3.12 – Comando de “general interrogation” (C_IC_NA_1)

Figura II.47 – Mensagem C_IC_NA_1

0	1	1	0	0	1	0	0	Identificador de Tipo
0	0	0	0	0	0	0	1	Qualificador Variável da Estrutura
Causa da Transmissão								
Endereço Comum da ASDU								
Endereço do Objeto de Informação								Objeto de Informação
UI8								QOI

QOI	8 bits
<0>	não utilizado
<1..19>	reservado
<20>	interrogação da estação (global)
<21>	interrogação do grupo 1
<22>	interrogação do grupo 2
<23>	interrogação do grupo 3
<24>	interrogação do grupo 4
<25>	interrogação do grupo 5
<26>	interrogação do grupo 6
<27>	interrogação do grupo 7
<28>	interrogação do grupo 8
<29>	interrogação do grupo 9
<30>	interrogação do grupo 10
<31>	interrogação do grupo 11
<32>	interrogação do grupo 12
<33>	interrogação do grupo 13
<34>	interrogação do grupo 14
<35>	interrogação do grupo 15
<36>	interrogação do grupo 16
<37..255>	reservado

II.3.6.3.13 – Comando de sincronismo do clock (C_CS_NA_1)

Figura II.48 – Mensagem C_CS_NA_1

<table>
<tr><td>0 1 1 0 0 1 1 1</td><td colspan="2">Identificador de Tipo</td></tr>
<tr><td>0 0 0 0 0 0 0 1</td><td colspan="2">Qualificador Variável da Estrutura</td></tr>
<tr><td colspan="2">Causa da Transmissão</td><td rowspan="2"></td></tr>
<tr><td colspan="2">Endereço Comum da ASDU</td></tr>
<tr><td colspan="2">Endereço do Objeto de Informação</td><td rowspan="2">Informação de tempo com 7 bytes
Data e horário do clock de milisegundos até anos</td><td rowspan="2">Objeto de Informação</td></tr>
<tr><td colspan="2">CP56Time2a</td></tr>
</table>

II.3.6.3.14 – Comando de teste (C_TS_NA_1)

Figura II.48 – Mensagem C_TS_NA_1

0	1	1	0	1	0	0	0	Identificador de Tipo
0	0	0	0	0	0	0	1	Qualificador Variável da Estrutura
Causa da Transmissão								
Endereço Comum da ASDU								
Endereço do Objeto de Informação								
1	0	1	0	1	0	1	0	Objeto de Informação FBP
0	1	0	1	0	1	0	1	

CAPÍTULO III – METODOLOGIA DE IMPLEMENTAÇÃO

Neste capítulo será descrito o funcionamento do protocolo de comunicação desenvolvido, explicando como foram utilizadas as diversas ferramentas, de forma estabelecer uma comunicação entre dois sistemas através das regras e normas exigidas pelo protocolo explicado no capítulo anterior.

Como descrito anteriormente, o modo de transmissão utilizado foi o não-balanceado. Neste modo, é possível identificar a presença de duas entidades bem definidas: o centro de supervisão e controle e a estação remota. A primeira é responsável por iniciar a transmissão de mensagens, solicitando informação e enviando comandos para a estação remota. Esta, por sua vez, somente pode transmitir mensagens de resposta às solicitações e pedidos da estação de controle. Desta forma, podemos caracterizar este sistema como um sistema cliente-servidor, onde a estação de controle desempenha o papel de cliente enquanto que a estação remota pode ser alocada no papel de servidor.

O programa foi desenvolvido para ser implementado em um centro de supervisão e controle, sendo assim, este será responsável pelo envio de comandos e pedidos à remota, que por sua vez ficará responsável por transmitir as respostas adequadas. Com o único objetivo de testar o correto funcionamento do programa foi também desenvolvido um modelo simplificado do processo utilizado por uma estação remota. Este, todavia, não será discutido, pois o foco estará centrado no funcionamento do processo da estação de controle, futuramente também chamado de cliente.

Como pode ser verificado nas seções iniciais do capítulo anterior, o protocolo descrito apresenta definições para a camada física, para a de link, para a camada de aplicação e camadas acima desta. O método de transmissão utilizado pela camada física é baseado em uma transmissão serial. De forma a facilitar a implementação, considerando que a realização de um link serial não está no escopo do programa apresentado, as mensagens encapsuladas no nível da camada de link foram transmitidas utilizando o protocolo TCP/IP. É importante ressaltar, que esta escolha não irá descaracterizar o protocolo desenvolvido, pois os métodos utilizados para a transmissão da mensagem são independentes da lógica e do funcionamento das camadas de link e de aplicação.

III.1 – Considerações gerais

De forma geral, o processo cliente tem início com o estabelecimento de uma conexão com o processo em execução na estação remota. Inicialmente, é criado um socket e é feita uma conexão com o processo remoto. Uma vez que esta foi estabelecida, o cliente dá início a uma função que pode ser caracterizada como uma máquina de estados. Num primeiro momento, esta envia um comando para a estação remota e em seguida espera pela resposta. Uma vez que a resposta é recebida, a estação de controle faz o tratamento destes dados e volta para o estado inicial de envio de mensagem, porém em uma nova condição que representa as condições necessárias para o envio da próxima mensagem.

Vale ressaltar que o programa é composto por dois arquivos fonte: tcpclient.c e lnk_trata_eventos.c. O primeiro apenas é responsável por criar o socket e estabelecer a conexão. O segundo arquivo contém a função responsável pela realização de todos os

processos necessários para estabelecer a troca de mensagens, segundo os critérios e normas exigidos pelo protocolo.

São utilizados ainda os arquivos: unp.h e lnk_layer.h. O primeiro contém as definições e estruturas importantes para a transmissão e o recebimento das mensagens, enquanto que a segundo possui as definições e estruturas características do protocolo.

III.2 – A função principal "main"

Conforme descrito anteriormente, primeiramente criamos um socket, definimos a porta e o IP de destino e realizamos a conexão com a estação remota.

Figura III.1 – Função "main"

```
sockfd = Socket(AF_INET, SOCK_STREAM, 0);

bzero(&servaddr, sizeof(servaddr));
servaddr.sin_family = AF_INET;
servaddr.sin_port = htons(9877);
Inet_pton(AF_INET, argv[1], &servaddr.sin_addr);

Connect(sockfd, (SA *) &servaddr, sizeof(servaddr));
printf("connected!\n");

// chama lnk_trata_eventos.c
Espera_e_trata_eventos(stdin, sockfd);
```

tcpclient.c

A função **Espera_e_trata_eventos** recebe como parâmetro o socket e é responsável por todos os processos referentes ao protocolo. Considerando que esta função possui um loop infinito, uma vez terminada sua execução o cliente também termina.

III.3 – A função "Espera_e_trata_eventos"

A função "**Espera_e_trata_eventos**" é composta por outras 14 funções, que junto com a rotina principal são responsáveis por executar diversas tarefas que compõem o protocolo. Estas funções serão descritas ao longo deste capítulo, à medida que forem sendo utilizadas pelo programa.

Figura III.2 – Funções contidas na "Espera_e_trata_eventos"

```
char *agora_iec();

void Inicializa_Var();

void Monta_msg_IEC_tam_fixo(unsigned char Ctrl,char Addr);

void Monta_msg_IEC_tam_var (unsigned char *Buffer, unsigned int Msg_len,unsigned char
Ctrl,char Addr);

int Transmite_msg_IEC (int sockfd, unsigned char *Buffer, unsigned int Msg_len );

void Zera_buffer(unsigned char *Buffer,int Tamanho);

int Recebe_primeiro_byte_IEC(int sockfd, unsigned char *buffer);

int Recebe_restante_IEC_fixo(int sockfd, unsigned char *buffer,char *Status_io);

int Recebe_restante_IEC_variavel(int sockfd, unsigned char *buffer,char *Status_io);

void Monta_byte_controle(unsigned char *Ctrl,char Func,char Status_ult_msg);

void Verifica_msg_IEC(int sockfd, unsigned char *Buffer_rx,char *Status_msg_rx,char
End_utr,char *Tipo_msg);

void Ocorreu_erro();

void Ocorreu_timeout();

int Trata_byte_controle (unsigned char Byte_ctl,char *Acd_c,char *Dfc_c,int *Function);

unsigned char CheckSum(const unsigned char *User_data, const unsigned short int Tam_len);
```

lnk_trata_eventos.c

III.3.1 – A função "Inicializa_Var"

Inicialmente, a função "**Espera_e_trata_eventos**" inicializa as variáveis importantes para definir as condições e o estado inicial do processo.

Como o modo de transmissão não-balanceado admite que uma estação primária controle mais de uma remota, o cliente também apresenta esta funcionalidade, e o número de remotas é definido nesta função. São definidos também os endereços de cada uma das remotas, endereços estes que serão utilizados para compor as mensagens transmitidas, assim como as condições de timeout e o estado inicial do processo. Segundo especifica a norma, a primeira mensagem a ser transmitida necessita ser uma mensagem de pedido de status do link, sendo assim, o estado inicial para todas as remotas é definido como sendo uma condição de envio desta mensagem.

Sem perda de generalidade, iremos descrever o processo para uma situação onde apenas possuímos uma remota conectada no link de transmissão, entendendo que a expansão do processo para mais de uma remota é imediata.

É importante também que tenhamos a informação do estado do bit FCB presente no byte de controle para que possamos saber qual será o próximo valor deste campo na próxima mensagem enviada. A variável **Fcb** mantém este registro.

Figura III.3 – Função "Inicializa_Var"

```
/*****************************************************************************/
void Inicializa_Var()
/*****************************************************************************/
{
    int i;

    // Define nome do arquivo de erro
    strcpy(Er,"error.log");

    // Estabelece numero de remotas (default=1)
    Num_remotas = DEF_NUM_REM;
    printf("Numero de Remotas: %d\n",Num_remotas);

    for (i = 0; i < Num_remotas; i++)
    {
            // Endereco das remotas (default remota 0=100)
            Address[i] = DEF_ADDR_REM_0;
            printf("Endereco da remota %d: %d\n",i+1,Address[i]);

            // Define numero maximo de tentativas
            Nmaxtent[i] = DEF_NUM_MAX_TENT;
            printf("Numero maximo de tentativas: %d\n",Nmaxtent[i]);
```

```
            // Define o estado inicial das remotas
            Estado[i]= ENVIA_REQUEST_STATUS_OF_LINK;

            Recebeu_msg_IEC_valida[i] = TRUE;

            Ocorreu_timeout_parcial = FALSE;

            Fcb[i] = 0;

            // Define link como fora. Ficara ON caso STATUS OF LINK e RESET LINK estejam OK
            Link[i] = OFF;

            Num_tent[i] = 0;

    }

}
```

lnk_trata_eventos.c

III.3.2 – A função "Monta_msg_IEC_tam_fixo"

Após a inicialização das variáveis de estado, o cliente está pronto para entrar na condição de envio da mensagem para a remota. Como visto anteriormente, o estado inicial do processo determina o envio de uma mensagem de pedido de status do link. De acordo com a evolução das trocas de mensagens entre as duas estações e em função das respostas recebidas da remota, a estação de controle irá conseqüentemente mudando sua condição de envio de mensagem. Por exemplo, supondo que a remota tenha respondido corretamente ao pedido de status do link enviado, a próxima condição da estação primária será a de envio de uma mensagem de reset do link. Para cada uma das condições de transmissão, mensagens apropriadas são construídas e enviadas, sempre obedecendo às regras estipuladas pela norma.

Usando como exemplo a primeira condição de transmissão, a estação primária irá, em primeiro lugar, montar a mensagem a ser transmitida, transmitindo-a em seguida e, posteriormente, definirá a condição de recebimento da mensagem de resposta.

O processo de montagem de uma mensagem de tamanho fixo é realizado pela função "**Monta_msg_IEC_tam_fixo"**. Esta recebe como parâmetros o byte de controle e o endereço

do link que é o endereço da remota. Uma vez recebidos estes parâmetros, esta função calcula o valor de byte de verificação ou checksum e posiciona os cinco bytes que compõem uma mensagem de tamanho fixo no buffer de transmissão para que sejam posteriormente enviados. O cliente já está pronto para a transmissão, porém antes de apresentarmos a função de envio será descrito como é feito o cálculo do byte de verificação.

Figura III.4 – Função "Monta_msg_IEC_tam_fixo"

```
/*****************************************************************************/
void Monta_msg_IEC_tam_fixo(unsigned char Ctrl,char Addr)
/*****************************************************************************/
// Esta rotina recebe os bytes de controle e endereco de utr
// do link layer, calcula o checksum da msg(para msg de tam fixo
// o checksum e calculado utilizando-se somente o byte de controle e o
// endereco da utr) e monta uma mensagem de tamanho fixo para
// ser enviada no buffer Msg_Tx_IEC.

{

  unsigned char Crc;
  unsigned char Buffer[2];

  // Header tamanho fixo
  MSGFIX *Msg_fix_ptr;

  Msg_fix_ptr=(MSGFIX *)Msg_Tx_IEC;
  //Msg_fix_ptr=(MSGFIX *)malloc(sizeof(MSGFIX));

  //Msg_fix_ptr=(MSGFIX *)Msg_Tx_IEC;

  // Copia o Start Byte - byte 0
  Msg_fix_ptr->start = BYTE_INICIO_MSG_TAM_FIXO_IEC;

  // Copia para a area comum os
  // bytes de controle e endereco

  // Copia byte de controle - byte 1
  Msg_fix_ptr->control = Ctrl;

  // Copia byte de endereco - byte 2
  Msg_fix_ptr->address = Addr;

  // Calcula o CheckSum
  Buffer[0] = Ctrl;
  Buffer[1] = Addr;
  Crc = (unsigned char)CheckSum(Buffer,2);

  // Copia o CheckSum - byte 3
  Msg_fix_ptr->checksum = Crc;

  // Copia o End Byte - byte 4
  Msg_fix_ptr->end = BYTE_FIM_MSG_IEC;

 // printf("\n Monta_msg_IEC_tam_fixo");
}
```

lnk_trata_eventos.c

III.3.3 – O byte de verificação ou checksum

Conforme discutido no capítulo anterior para o formato de frame FT 1.2, a norma define a presença de um byte de verificação, tanto nas mensagens de tamanho variável quanto nas mensagens de tamanho fixo, sendo que em ambos está presente no penúltimo byte da mensagem transmitida.

Sua presença se dá de forma a manter a confiabilidade dos dados transmitidos, e pode ser calculado como sendo a soma dos bytes contidos no "user data", segundo mostrado na seção **II.2.2.1.2**. Esta soma é feita módulo 256, ou seja, no byte destinado ao checksum é armazenado o resto da divisão da soma dos bytes do "user data" por 256.

Para o caso da mensagem de tamanho fixo, o checksum é a soma do byte de controle com o byte de endereço. Já para o caso da mensagem de tamanho variável, somamos também os bytes que compõem a ASDU segundo descrito na seção **II.2.6**.

A função responsável pelo cálculo do byte de verificação é a função "**CheckSum**".

Figura III.5 – Função "CheckSum"

```
/*****************************************************************************/
// CheckSum para o formato FT 1.2 - Arithmetic sum modulo 256
/*****************************************************************************/
unsigned char CheckSum(const unsigned char *User_data, const unsigned short int Tam_len)
// Esta rotina retorna o valor da soma modulo 256 dos tam_len bytes a partir
// do endereco user_data.
// Este valor deve estar presente no byte de checksum da mensagem, que se
// encontra entre o ultimo byte do user_data e o byte de fim da mensagem.
// Ele deve ser checado ao se receber uma mensagem IEC e inserido ao se enviar
// uma mensagem IEC
{
  int i,Sum=0;
  for (i=0;i<Tam_len;i++) {
    Sum+=(int)User_data[i];
  }
  return (unsigned char)Sum%256;
}
```

lnk_trata_eventos.c

III.3.4 – A função de transmissão "Transmite_msg_IEC"

O próximo passo após a formação da mensagem a ser transmitida é a própria transmissão da mesma, e isto é feito por esta função que é extremamente simples.

Esta função recebe três parâmetros: o socket, um ponteiro para o buffer de transmissão e o tamanho da mensagem a ser transmitida. Esses três valores são utilizados na função **Writen** que é uma adaptação proposta por *Stevens* à função primitiva **write**. Diferente da segunda função, a primeira se certifica que o total de bytes solicitados foi realmente enviado. Isto ocorre, pois por limitações de buffer, às vezes é necessária mais de uma operação de I/O para o envio por completo da mensagem. Devido à característica do protocolo, que limita o tamanho da mensagem transmitida em até 261 bytes, é provável que não seja necessário a realização de mais de uma operação de I/O, porém esta adaptação será utilizada para garantir a integridade e para evitar possíveis problemas.

Figura III.6 – Função "Transmite_msg_IEC"

```
/*****************************************************************************/
int Transmite_msg_IEC (int sockfd, unsigned char *Buffer, unsigned int Msg_len )
/*****************************************************************************/
// Esta rotina recebe um buffer com tamanho Msg_len e o envia
// para a utr IEC

{


    Writen(sockfd,Buffer,Msg_len);


    /*for (i=0;i<Msg_len;i++)
    {
      printf("%d) %Xh \n",i,Buffer[i]);
    }
      printf("\n");
    */
    return 0;

}
```

lnk_trata_eventos.c

III.3.5 – O estado de espera pela mensagem de resposta

Uma vez enviada a mensagem para a remota, o cliente necessita agora esperar pela mensagem de resposta para que possam ser tomadas as medidas necessárias, porém antes são definidas duas condições. A primeira define qual será o estado de espera, ou seja, após o recebimento da mensagem de resposta, o cliente, de acordo com o estado de espera, irá promover o tratamento adequado da mensagem recebida. A segunda irá definir qual será o estado do cliente caso ocorra timeout na mensagem enviada.

Uma vez definidas essas duas condições, o processo executa a função **Select** que espera pela ocorrência de uma das duas condições: recebimento da mensagem ou timeout. Iremos primeiramente descrever as medidas tomadas em caso de timeout, para em seguida continuarmos a explicação do processo.

III.3.6 – Ocorrência de timeout

Caso a mensagem enviada não tenha sido recebida pela remota, ou caso a resposta não tenha sido recebida pela estação de controle, o programa entra na condição de timeout. O tempo estipulado para o timeout não é definido pela norma e deve ser acordado entre as partes envolvidas. Levando em consideração, que o tempo aproximado entre a transmissão de duas mensagens consecutivas foi definido para este programa como sendo de um segundo, utilizamos um tempo de timeout de cinco segundos por considerar este período de tempo suficientemente grande para a transmissão e o processamento dos dados.

Uma vez ocorrido o timeout, o cliente entra no estado de timeout correspondente à última mensagem enviada. Neste estado, é executada a função "**Ocorreu_timeout"** que é responsável pelo controle do número de mensagens retransmitidas e pelo estado do link. Num primeiro momento quando ocorre um timeout o cliente tenta retransmitir a última mensagem enviada. Esse processo é repetido um número determinado de vezes, número esse variável de acordo com a implementação do programa, até que, estourado o limite de tentativas, o cliente considera o link como estando fora de serviço e determina que o próximo estado de envio de mensagens será o estado inicial, ou seja, o estado no qual a mensagem transmitida é a de pedido de status do link. A partir de então se inicia novamente o processo de verificação do status do link.

Caso ocorra timeout em uma determinada mensagem, porém a mensagem seguinte seja respondida adequadamente, o contador de timeout é zerado.

Figura III.7 – Função "Ocorreu_timeout"

```
/*****************************************************************************/
void Ocorreu_timeout()
/*****************************************************************************/
// Esta rotina contabiliza os timeouts ocorridos
// se o numero for maior que o permitido, derruba
// o link

{
  // Se nao estourou o limite de timeouts
  if(Num_tent[Remota] < Nmaxtent[Remota])
  {
    Num_tent[Remota]++;
  }
  else                  // Estourou o limite de timeouts
  {
    Num_tent[Remota]= 0;
    Fcb[Remota] = 0;

                        // Se o link estava dentro, derruba
                        // e avisa a aplicacao
    if(Link[Remota]== ON)
    {
      Tam_msg_Tx_APPL = 0;
      Envia_msg_APPL(Msg_Tx_APPL,LINK_FORA,Tam_msg_Tx_APPL,Remota);
      Link[Remota] = OFF;
    }
  }
}
```

lnk_trata_eventos.c

III.3.7 – As condições de recebimento de mensagens

Caso a mensagem tenha sido recebida corretamente, o cliente prossegue com o processo de tratamento dos dados recebidos pela remota. Em primeiro lugar, o cliente executa a função "**Verifica_msg_IEC"** que irá verificar a mensagem recebida quanto a sua consistência. Caso um erro tenha sido detectado, este será adequadamente tratado pela função "**Ocorreu_erro"**. Por outro lado, se a mensagem recebida estiver correta, inicialmente será analisado o byte de controle para que em função dos dados contidos nele possa ser analisada a coerência dos dados recebidos com relação ao pedido efetuado. Por exemplo, espera-se que uma mensagem de pedido de status do link tenha como resposta uma mensagem de status do link. Caso a mensagem recebida seja um ACK isto deverá ser tratado como um erro, e a última mensagem enviada pela estação de controle deverá ser retransmitida.

Se a resposta esperada for recebida, o cliente passará para um novo estado de envio de mensagens. No exemplo anterior, o recebimento de uma mensagem de status do link pela estação primária fará com que esta entre no estado de envio da mensagem de reset do link.

A seguir, detalharemos as funções **"Verifica_msg_IEC"** e **"Ocorreu_erro"** para posteriormente continuarmos com a explicação da função principal.

III.3.8 – A função "Verifica_msg_IEC"

Esta rotina recebe três parâmetros e retorna dois. Os parâmetros recebidos são: o socket, o buffer de recebimento da mensagem de resposta enviada pela remota e o endereço da remota.

Os dois valores retornados são: o status da mensagem recebida (com sucesso ou sem sucesso) e o tipo de mensagem recebida (tamanho fixo, variável ou caractere único).

Como a estação primária não pode determinar a priori qual o tamanho da mensagem a ser recebida, o processo de recebimento dos dados pode ser feito em uma, duas ou até três etapas. Como dito anteriormente, após o envio da mensagem para a remota, o cliente fica a espera de uma resposta. Uma vez recebida, o cliente lê apenas o primeiro byte desta mensagem. A partir deste byte o cliente poderá determinar se a mensagem recebida é de tamanho fixo, variável ou um caractere único. Até a execução da função "**Verifica_msg_IEC**", somente o primeiro byte da resposta foi lido. Esta função ficará responsável por determinar o tipo de mensagem, irá fazer mais um I/O caso a mensagem recebida seja de tamanho fixo, ou mais dois se for de tamanho variável, além de efetuar as verificações necessárias na mensagem recebida.

Dentre as verificações realizadas, esta função verifica o valor do byte de início e de fim, calcula o checksum dos dados recebidos e compara com o valor de checksum recebido, verifica se o endereço da remota de origem corresponde com o endereço esperado e para a mensagem de tamanho variável compara os dois bytes de tamanho da mensagem e verifica se este valor está condizente com o tamanho real da mensagem.

III.3.9 – A função "Ocorreu_erro"

Mostramos que a função descrita anteriormente realiza uma série de verificações quanto à consistência da mensagem recebida. Caso a mensagem recebida esteja consistente, isto é indicado através da variável **Status_msg_IEC** que é retornada pela função anterior. A função principal (**Espera_e_trata_eventos**) verifica o valor desta variável. Caso tenha ocorrido

algum erro, a função **"Ocorreu_erro"** é executada e, assim com a função **"Ocorreu_timeout"**, irá tratar o problema adequadamente.

A função **"Ocorreu_erro"** executa os mesmos procedimentos que a função **"Ocorreu timeout"**, ou seja, irá incrementar os contadores que indicam o número de tentativas sem sucesso e irá declarar o link como estando fora de serviço caso ocorram erros repetitivos.

III.3.10 – A função "Trata_byte_controle"

Caso a mensagem tenha sido recebida com sucesso, o próximo passo é a execução da função **"Trata_byte_controle"** que recebe um parâmetro indicando qual o tipo de mensagem (tamanho fixo, variável ou caractere único), e retorna três informações importantes presentes no byte de controle, discutido na seção **I.2.2.2**, que necessitam serem analisadas detalhadamente para o correto processamento da mensagem recebida. Estas informações são: o bit ACD, o bit DFC e os quatro bits que compõem a função de serviço.

O bit ACD irá indicar à estação de controle que a remota possui dados de classe 1 prontos para transmissão. A partir desta informação, a estação de controle poderá então definir qual será o próximo estado de envio de mensagens. Caso existam dados de classe 1 para serem transmitidos, a estação de controle irá, na sua próxima mensagem, solicitar por dados desta classe ou, caso não existam (indicado pelo bit ACD igual a 0), irá solicitar dados de classe 2.

O bit DFC permite informar à estação de controle que futuras mensagens enviadas à remota poderão ser perdidas devido a overflow do buffer de recebimento desta. Este campo permite que a estação primária espere um tempo maior para o envio da próxima mensagem.

Por último, a informação do código da função de serviço irá identificar qual o tipo de mensagem de link retornada à estação de controle pela remota. Por exemplo, uma mensagem enviada da estação de controle para a remota com função de serviço igual a 9 (pedido de status do link) deverá ser respondida com uma mensagem cuja função de serviço seja igual a 11 (status do link). Desta forma, caso a função de serviço recebida não seja condizente com a função enviada, a resposta será tratada como um erro, e a mensagem anterior será retransmitida.

Figura III.8 – Função "Trata_byte_controle"

```
/*****************************************************************************/
int Trata_byte_controle(unsigned char Tipo_ctl,char *Acd_c,char *Dfc_c,int *Function)
/*****************************************************************************/
// Esta rotina recebe o tipo da msg
// enviado pela remota e o divide em 3:
// Acd:Access_demand,Dfc:Data flow control e Function:Funcao
{
  unsigned char Byte_ctl;

  switch(Tipo_ctl)
  {
    case TAM_FIXO:
    {
      Byte_ctl = Msg_Rx_IEC[1];
      break;
    }
    case TAM_VAR:
    {
      Byte_ctl = Msg_Rx_IEC[4];
      break;
    }
    case TAM_SHORT:
    {
      Byte_ctl = Msg_Rx_IEC[1];
      break;
    }
    default:
    {
      Byte_ctl = 0;
      return -1;
    }
  }

  *Function = (int)( Byte_ctl & 0x0F);
  *Dfc_c    = (int)((Byte_ctl & 0x10) >> 4);
  *Acd_c    = (int)((Byte_ctl & 0x20) >> 5);

  printf("Function: %X\n",*Function);
  printf("Dfc: %X\n",*Dfc_c);
  printf("Acd: %X\n",*Acd_c);

  return 0;
}
```

lnk_trata_eventos.c

III.3.11 – As mensagens de tamanho variável

Basicamente, já foram descritos todos os processos que compõem a troca de mensagens entre a estação primária e a secundária. Nos resta agora analisar duas situações especiais que não foram abordadas até o presente momento, ou seja, o envio e o recebimento de mensagens de tamanho variável e o envio de mensagens de comando. Até agora, todas as mensagens enviadas e recebidas se enquadravam no tipo de tamanho fixo. Após a troca inicial de mensagens, onde são verificados as condições do link e os estados das camadas de aplicação e de link das duas estações, a estação de controle dá início ao envio de pedidos de dados de classe 1 e 2 à remota. A partir de então, a estação de controle deve estar preparada para o recebimento de mensagens de tamanho fixo, e uma vez recebendo este tipo de mensagem, estas são encaminhadas para a camada de aplicação que será responsável por depurar e tratar os dados contidos na ASDU.

III.3.12 – A camada de aplicação

A camada de aplicação será responsável por analisar os dados recebidos de forma que o usuário possa ter o controle e a visualização das informações presentes na remota. A partir destes dados o usuário poderá tomar as medidas cabíveis e pertinentes para a supervisão da estação remota. A camada de aplicação basicamente se apresenta como uma interface entre o usuário e a camada de link. Dentre as funções da camada de aplicação podemos citar: o tratamento dos dados de nível de aplicação recebidos da estação remota, armazenamento da informação recebida de forma apropriada em bancos de dados, envio dos dados relevantes para aplicativos gerenciais que serão responsáveis pela visualização dos dados e pela interface de supervisão, tomada de decisões quanto ao envio de comandos de teste, “general

interrogation" ou de sincronismo do clock, entre outros. A seguir, iremos descrever resumidamente alguns procedimentos executados pela camada de aplicação no recebimento de dados da remota e no envio de comando para esta.

III.3.12.1 – Recebendo ASDUs da remota

Ao receber uma mensagem de tamanho variável da remota, a camada de link encaminha esta mensagem para a camada de aplicação que por sua vez ficará responsável pelo tratamento dos dados recebidos e pelo envio das informações importantes para os aplicativos do usuário.

Ao receber uma ASDU, a camada de aplicação primeiramente identifica qual o tipo de mensagem recebida. Se esta mensagem for uma informação que somente interessa a esta camada, como uma resposta a um comando de teste ou de sincronismo de clock, a mensagem será devidamente tratada, e os devidos contadores e temporizadores serão atualizados. Se a informação recebida for o estado ou o valor de algum elemento da remota, como uma chave digital ou o valor de uma linha de tensão, estes dados serão encaminhados para os processos responsáveis pelo tratamento específico de cada tipo de dado.

III.3.12.2 – Enviando comandos para a remota

Outra situação na qual é necessário lidar com mensagens de tamanho variável é quando a camada de aplicação envia comandos e pedidos especiais para a remota. Algumas mensagens como "general interrogation", sincronismo de clock e teste de link são exemplos de mensagens de tamanho variável. Quando a camada aplicação decide por enviar alguma destas mensagens, esta envia para a camada de link a ASDU em questão. Esta camada por sua vez

irá encapsular a ASDU com os bytes restantes que compõem uma mensagem de tamanho variável enviando em seguida a mensagem para a remota. A ASDU recebida da camada de aplicação pela camada de link é encapsulada através da função **"Monta_msg_IEC_tam_var"**. Esta função desempenha o mesmo papel da função "**Monta_msg_IEC_tam_fixo"** descrita anteriormente e por este motivo não será explicada mais detalhadamente. Vale ressaltar, que apesar da tarefa realizada pelas duas funções ser a mesma, ou seja, encapsular a mensagem com as informações da camada de link, ambas estruturam os dados de forma diferente, pois os formatos das mensagens de tamanho fixo e variável são diferentes.

CAPÍTULO IV – TESTES

Neste capítulo, com o objetivo de ilustrar o funcionamento do programa, será apresentada uma seqüência de testes, a qual foi submetido o protocolo em questão, mostrando a dinâmica das trocas de mensagens, assim como os tipos de mensagens trocadas entre os dois sistemas.

O protocolo inicialmente desenvolvido em Furnas foi testado, em um primeiro momento, por um aplicativo comercial denominado ASE2000. Este programa pode assumir o papel de uma estação primária, assim como o de uma remota. Desta forma, conectou-se o protocolo ao aplicativo com o objetivo de realizar alguns testes preliminares. O ASE2000, quando operando como uma remota, é capaz de desempenhar tarefas típicas de uma estação secundária, como gerar eventos e responder apropriadamente às mensagens enviadas. Em um segundo momento, conectou-se o protocolo a uma subestação e foram feitos testes reais em várias situações diferentes.

Os testes apresentados neste projeto foram realizados no ambiente operacional Linux na versão 7.3 e o protocolo foi testado em duas situações diferentes. Na primeira delas o cliente e o servidor estavam localizados no mesmo computador e a conexão foi feita através do endereço IP local (127.0.0.1). Na segunda situação o cliente foi executado em um computador enquanto que o servidor ficou localizado em computador remoto. Em ambos os casos os testes funcionaram corretamente conforme o esperado.

IV.1 – Ativação da remota

Em um primeiro momento será ilustrada a situação onde o processo da estação de controle está ativo e em operação enquanto que a remota está fora de serviço. Poderá ser verificado que a estação de controle enviará um pedido de status do link. Como esta mensagem não foi retornada pela remota, o que pode ser verificado pela ocorrência de timeout indicada, serão enviadas repetidas mensagens de pedido de status do link até obter uma resposta de status do link. A seguir, a estação primária enviará um pedido de reset do link que a remota responde com um ACK. A partir deste momento, já podem ser enviadas as mensagens de pedido de dados.

Figura IV.1 – Ativação da remota

```
Esperando 1 seg...
ENVIA_REQUEST_STATUS_OF_LINK
0) 10 h
1) 49 h
2) 64 h
3) AD h
4) 16 h

Ocorreu timeout
Esperando 1 seg...
ENVIA_REQUEST_STATUS_OF_LINK
0) 10 h
1) 49 h
2) 64 h
3) AD h
4) 16 h

Ocorreu timeout
Esperando 1 seg...
ENVIA_REQUEST_STATUS_OF_LINK
0) 10 h
1) 49 h
2) 64 h
3) AD h
4) 16 h

MENSAGEM_RECEBIDA
0) 10 h
1) B h
2) 64 h
3) 6F h
4) 16 h

Esperando 1 seg...
ENVIA_RESET_OF_REMOTE_LINK
0) 10 h
1) 40 h
2) 64 h
3) A4 h
4) 16 h
```

```
MENSAGEM_RECEBIDA
0) 10 h
1) 0 h
2) 64 h
3) 64 h
4) 16 h

Esperando 1 seg...
ENVIA_REQUEST_USERDATA_CLASSE2
0) 10 h
1) 7B h
2) 64 h
3) DF h
4) 16 h

MENSAGEM_RECEBIDA
0) 10 h
1) 9 h
2) 64 h
3) 6D h
4) 16 h
```

IV.2 – Remota fora de serviço

Nesta situação, será mostrado o comportamento do sistema, particularmente a estação de controle, considerando que a remota a partir de um certo momento não mais responde às solicitações da estação de controle. Esta situação procura ilustrar a troca de mensagens para o caso de uma possível queda no link de transmissão, ou uma eventual inicialização da estação remota.

Nota-se, conforme ilustrado abaixo, que uma vez não recebida a resposta para uma mensagem enviada pela estação de controle, esta irá retransmitir a mesma mensagem enviada anteriormente após um período de timeout. Este processo irá se repetir até que o contador de timeout alcance o seu valor limite, quando então, serão posteriormente enviadas mensagens de pedido do status do link até que este seja restabelecido novamente.

Figura IV.2 – Remota fora de serviço

```
Esperando 1 seg...
ENVIA_REQUEST_USERDATA_CLASSE2
0) 10 h
1) 7B h
2) 64 h
3) DF h
4) 16 h

MENSAGEM_RECEBIDA
0) 10 h
1) 9 h
2) 64 h
3) 6D h
4) 16 h

Esperando 1 seg...
ENVIA_REQUEST_USERDATA_CLASSE2
0) 10 h
1) 5B h
2) 64 h
3) BF h
4) 16 h

MENSAGEM_RECEBIDA
0) 10 h
1) 9 h
2) 64 h
3) 6D h
4) 16 h

Esperando 1 seg...
ENVIA_REQUEST_USERDATA_CLASSE2
0) 10 h
1) 7B h
2) 64 h
3) DF h
4) 16 h

Ocorreu timeout
Esperando 1 seg...
ENVIA_REQUEST_USERDATA_CLASSE2
0) 10 h
1) 7B h
2) 64 h
3) DF h
4) 16 h

...

Ocorreu timeout
Esperando 1 seg...
ENVIA_REQUEST_STATUS_OF_LINK
0) 10 h
1) 49 h
2) 64 h
3) AD h
4) 16 h

Ocorreu timeout
Esperando 1 seg...
ENVIA_REQUEST_STATUS_OF_LINK
0) 10 h
1) 49 h
2) 64 h
3) AD h
4) 16 h
```

IV.3 – Aquisição de dados

A aquisição de dados representa um processo importante na troca de mensagens entre um centro de supervisão e controle e uma estação remota. Serão mostrados nesta seção os diferentes tipos de mensagens correspondentes à aquisição de dados analógicos e digitais.

Em primeiro lugar é enviado um pedido de dados de classe 2. Este pedido é respondido com o estado de um elemento digital simples. Podemos verificar que a remota sinalizou, através do byte de controle, que existem dados de classe 1 que necessitam serem transmitidos. Na sua próxima mensagem, a estação primária envia um pedido por dados de classe 1, pedido este que é respondido com o valor de um elemento analógico. Como a remota não indicou que necessitava transmitir informações de classe 1, a próxima pergunta buscará dados de classe 2. Como resposta, a estação de controle receberá o estado de um elemento digital duplo.

Figura IV.3 – Aquisição de dados

```
Esperando 1 seg...
ENVIA_REQUEST_USERDATA_CLASSE2
0) 10 h
1) 7B h
2) 64 h
3) DF h
4) 16 h

MENSAGEM_RECEBIDA
0) 68 h
1) A h
2) A h
3) 68 h
4) 28 h
5) 64 h
6) 1 h
7) 1 h
8) 5 h
9) 0 h
10) 0 h
11) 1 h
12) 0 h
13) 1 h
14) 95 h
15) 16 h

Esperando 1 seg...
ENVIA_REQUEST_USERDATA_CLASSE1
0) 10 h
1) 5A h
2) 64 h
```

```
3) BE h
4) 16 h

MENSAGEM_RECEBIDA
0) 68 h
1) C h
2) C h
3) 68 h
4) 8 h
5) 64 h
6) 9 h
7) 1 h
8) 5 h
9) 0 h
10) 0 h
11) 2 h
12) 0 h
13) 0 h
14) 1 h
15) 0 h
16) 7E h
17) 16 h

Esperando 1 seg...
ENVIA_REQUEST_USERDATA_CLASSE2
0) 10 h
1) 7B h
2) 64 h
3) DF h
4) 16 h

MENSAGEM_RECEBIDA
0) 68 h
1) A h
2) A h
3) 68 h
4) 8 h
5) 64 h
6) 3 h
7) 1 h
8) 5 h
9) 0 h
10) 0 h
11) 3 h
12) 0 h
13) 2 h
14) 7A h
15) 16 h
```

IV.4 – Envio de "general interrogation"

Nesta seção serão exemplificadas as mensagens trocadas quando o centro de supervisão e controle envia uma mensagem de "general interrogation". Este tipo de mensagem é enviado quando se deseja atualizar a base de dados do cliente com as informações de todos os elementos de informação presentes no servidor.

A estação de controle envia um comando de interrogação indicando que a causa de transmissão é ativação do serviço. A remota confirma o correto recebimento do pedido retornando a mesma mensagem para o solicitante, porém com a causa de transmissão igual à confirmação de ativação de serviço. A partir deste momento, a remota dará início ao envio de todos os elementos que estão contidos no seu banco de dados e que são de interesse para a estação primária.

Após o envio de todos os elementos, a remota indica que terminou o envio destes através de uma mensagem de interrogação com a causa de transmissão igual à terminação de ativação do serviço. Vale ressaltar, que enquanto a estação remota está enviando os elementos em resposta a esta solicitação em particular, a causa de transmissão deste é igual a: interrogada pelo comando de interrogação. Isto ocorre, para diferenciar as mensagens de resposta ao comando de interrogação, das mensagens de resposta a uma solicitação de dados de classe 1 ou 2, que possuem causa de transmissão igual a: solicitado.

Figura IV.4 – Envio de "general interrogation"

```
Esperando 1 seg...
ENVIA_INTERROGATION_COMMAND
0) 68 h
1) A h
2) A h
3) 68 h
4) 5B h
5) 64 h
6) 64 h
7) 1 h
8) 6 h
9) 0 h
10) 0 h
11) 0 h
12) 0 h
13) 14 h
14) 3E h
15) 16 h

MENSAGEM_RECEBIDA
0) 68 h
1) A h
2) A h
3) 68 h
4) 8 h
5) 64 h
6) 64 h
7) 1 h
```

```
8) 7 h
9) 0 h
10) 0 h
11) 0 h
12) 0 h
13) 14 h
14) EC h
15) 16 h

Esperando 1 seg...
ENVIA_REQUEST_USERDATA_CLASSE2
0) 10 h
1) 7B h
2) 64 h
3) DF h
4) 16 h

MENSAGEM_RECEBIDA
0) 68 h
1) A h
2) A h
3) 68 h
4) 8 h
5) 64 h
6) 1 h
7) 1 h
8) 14 h
9) 0 h
10) 0 h
11) 0 h
12) 0 h
13) 0 h
14) 82 h
15) 16 h

...

Esperando 1 seg...
ENVIA_REQUEST_USERDATA_CLASSE2
0) 10 h
1) 7B h
2) 64 h
3) DF h
4) 16 h

MENSAGEM_RECEBIDA
0) 68 h
1) A h
2) A h
3) 68 h
4) 8 h
5) 64 h
6) 64 h
7) 1 h
8) A h
9) 0 h
10) 0 h
11) 0 h
12) 0 h
13) 14 h
14) EF h
15) 16 h
```

IV.5 – Envio de sincronismo de clock

A seguir, serão mostrados os procedimentos referentes ao envio de um comando de sincronismo do clock para a remota. Verificar-se-ão quais são as mensagens que compõem este procedimento.

A mensagem de sincronismo de clock enviada pela estação de controle é respondida com uma mensagem igual, porém com causa de transmissão diferente. As causas ativação e confirmação de ativação são utilizadas respectivamente.

Figura IV.5 – Envio de sincronismo do clock

```
Esperando 1 seg...
ENVIA_CLOCK_SYNCHRONIZATION
0) 68 h
1) 10 h
2) 10 h
3) 68 h
4) 5A h
5) 64 h
6) 67 h
7) 1 h
8) 6 h
9) 0 h
10) 0 h
11) 0 h
12) 0 h
13) 0 h
14) 0 h
15) 0 h
16) 8 h
17) 81 h
18) A h
19) 2 h
20) C1 h
21) 16 h

MENSAGEM_RECEBIDA
0) 68 h
1) 10 h
2) 10 h
3) 68 h
4) 8 h
5) 64 h
6) 67 h
7) 1 h
8) 7 h
9) 0 h
10) 0 h
11) 0 h
12) 0 h
13) 0 h
14) 0 h
15) 0 h
16) 8 h
```

```
17) 81 h
18) A h
19) 2 h
20) C1 h
21) 16 h
```

IV.6 – Envio de comandos

Finalmente, apresentaremos como se dá o envio de comandos do centro de supervisão e controle para a remota e como estão estruturadas essas mensagens.

Ilustramos o envio de um comando direto, respondido por uma mensagem igual porém com causa de transmissão diferente. Após a execução do comando pela remota, esta irá indicar este fato através do envio de uma mensagem de envio de comando com a causa de transmissão igual a: terminação da ativação.

Figura IV.6 – Envio de comandos

```
Esperando 1 seg...
ENVIA_SINGLE_COMMAND
0) 68 h
1) A h
2) A h
3) 68 h
4) 5B h
5) 64 h
6) 2D h
7) 1 h
8) 6 h
9) 0 h
10) 0 h
11) 0 h
12) 0 h
13) 5 h
14) F8 h
15) 16 h

MENSAGEM_RECEBIDA
0) 68 h
1) A h
2) A h
3) 68 h
4) 8 h
5) 64 h
6) 2D h
7) 1 h
8) 7 h
9) 0 h
10) 0 h
11) 0 h
12) 0 h
```

```
13) 5 h
14) A6 h
15) 16 h

Esperando 1 seg...
ENVIA_REQUEST_USERDATA_CLASSE2
0) 10 h
1) 7B h
2) 64 h
3) DF h
4) 16 h

MENSAGEM_RECEBIDA
0) 68 h
1) A h
2) A h
3) 68 h
4) 8 h
5) 64 h
6) 2D h
7) 1 h
8) A h
9) 0 h
10) 0 h
11) 0 h
12) 0 h
13) 5 h
14) A9 h
15) 16 h
```

CAPÍTULO V – CONCLUSÃO

Inicialmente, começamos descrevendo as camadas que compõem um protocolo de comunicação através do modelo OSI e posicionamos este protocolo dentro deste modelo. A seguir, detalhamos cada uma das normas que compõem o protocolo desenvolvido com o objetivo de criar o embasamento teórico necessário para o correto entendimento dos processos de comunicação envolvidos.

A primeira norma descrita discorreu sobre os diferentes formatos ou tipos de frames de transmissão, estruturas estas que serão importantes para definir o tipo de encapsulamento dos dados provenientes das camadas superiores assim como para caracterizar o tipo de integridade garantida aos dados transmitidos. A segunda norma buscou detalhar os campos que compõem os frames de transmissão apresentados anteriormente além de definir funções características da camada de link. A terceira norma apresentada procurou esboçar de forma geral as estruturas pertencentes à camada de aplicação, assim como regras para a estruturação dos dados neste nível. A quarta camada apresentada definiu regras sobre os tipos de elementos de dados utilizados nas trocas de mensagens. Estas regras serão utilizadas posteriormente na formação das estruturas do nível de aplicação. A quinta norma estabeleceu algumas funções básicas também no nível de aplicação para a realização de tarefas de telemetria, supervisão e controle. Finalmente, a última norma definiu e detalhou as mensagens utilizadas na realização das tarefas de aquisição de dados, envio de comandos, entre outras funções.

Uma vez garantidos os pré-requisitos necessários ao entendimento dos processos que regem a troca de mensagens entre os sistemas foi apresentada uma implementação para o protocolo de comunicação descrito. A descrição e a implementação deste buscou realizar e concretizar os conceitos teóricos discutidos no decorrer da norma.

REFERÊNCIAS BIBLIOGRÁFICAS

STEVENS, W. R. “UNIX Network Programming”. Singapore: Pearson Education, Inc. 1999

International Standard, IEC 870-5-1 – Telecontrol equipment and systems, Part 5, Section 1: Transmission frame formats.

International Standard, IEC 870-5-2 – Telecontrol equipment and systems, Part 5, Section 2: Link transmission procedures.

International Standard, IEC 870-5-3 – Telecontrol equipment and systems, Part 5, Section 3: General structure of application data.

International Standard, IEC 870-5-4 – Telecontrol equipment and systems, Part 5, Section 4: Definition and coding of application information elements.

International Standard, IEC 870-5-5 – Telecontrol equipment and systems, Part 5, Section 5: Basic application functions.

International Standard, IEC 870-5-101 – Telecontrol equipment and systems, Part 5, Section 101: Companion standard for basic telecontrol tasks.

Norwegian IEC 870-5-101 User Conventions.