

自律分散プロトコル仕様書

R 3.0

MSTC/JOP 1101-1999/09/30

平成11年8月10日

財団法人 製造科学技術センター
F A オープン推進協議会 分散型製造システム専門委員会

財団法人 製造科学技術センター

F A オープン推進協議会
分散型製造システム専門委員会

〒105-0002 東京都港区愛宕 1-2-2 第9森ビル 7階

電話：(03)5472-2561

FAX：(03)5472-2567

H P：http://www.mstc.or.jp/

自律分散プロトコル仕様書 R2.0^{*1} (暫定版)

自律分散プロトコル仕様書 R3.0^{*2}

第1版

© F A オープン推進協議会

禁無断転載

本仕様書が規定するプロトコル仕様には、第3者より
F A オープン推進協議会に対して無償の通常実施権を許諾
された特許が含まれています。本仕様書の利用者は本文の
まえがきに記載された事項を遵守下さい。

^{*1} R2.0：R1.0(1998年12月7日発行)に「付属書Ⅰ．連想配列プロトコル」を追加。

^{*2} R3.0：R2.0(1999年3月19日発行)の構成を全面表記改訂。(仕様はR2.0と同一)

目次

まえがき	1
0章 序文	2
1章 適用範囲	2
2章 プロトコルの前提	3
2.1 前提となる構成	3
2.2 前提となる通信プロトコル	4
3章 用語の定義	5
3.1 ADP (Autonomous Decentralized Protocol)	5
3.2 機器	5
3.3 LANセグメント	5
3.4 論理ノード	5
3.5 データフィールド (DF)	6
3.5.1 ローカルデータフィールド	7
3.5.2 リモートデータフィールド	7
3.6 ドメイン	8
3.7 マルチキャストグループ	8
3.8 ブロードキャスト	9
3.9 マルチキャスト通信	9
3.10 1対1通信	9
3.11 生存信号	9
3.12 障害情報	9
3.13 AP(Application Program)	9
3.14 PDU(Protocol Data Unit)	10
3.15 メッセージ	10
3.16 トランザクション	10
3.16.1 ユーザトランザクション	10
3.16.2 システムトランザクション	11
3.17 実装者	11
3.18 ユーザ	11
4章 プロトコルの機能	12
4.1 クラス-Base-1 (マルチキャスト通信)	12
4.1.1 機能	12
4.1.2 マルチキャストグループの管理方法	13
4.2 クラス-Base-2 (生存信号送信)	16
4.2.1 機能	16
4.2.2 生存信号送信方式	16
4.2.3 生存信号によるノードの生死判定	18
4.2.4 UDPポート番号の割り当て方法	18
4.3 クラス-Opt-2-a (障害情報送出)	20
4.3.1 機能	20
4.3.2 障害情報	20

4.4	クラス-Opt-3 (1 対 1 通信)	22
4.4.1	機能	22
4.4.2	1 対 1 通信用TCPコネクション管理	23
4.5	テスト支援	25
4.5.1	機能	25
4.5.2	メッセージモード	25
4.5.3	ノードモード	25
4.5.4	メッセージ送受信制御	25
4.5.5	マルチキャスト受信ポートの割当	26
4.5.6	テスト支援機能処理	27
4.6	メッセージ優先制御	28
4.6.1	機能	28
4.6.2	メッセージ優先レベル	28
4.6.3	メッセージ優先制御処理	28
4.7	メッセージ通番管理	30
4.7.1	機能	30
4.7.2	メッセージ送信通番	30
4.7.3	メッセージ送信通番バージョン番号	30
4.7.4	メッセージ通番管理処理	31
4.8	メッセージの分割・組み立て	32
4.8.1	機能	32
4.8.2	メッセージ分割・組み立てに使用する情報	32
4.9	連想配列プロトコル	33
4.9.1	メッセージ構造	33
4.9.2	構造の表現	34
4.9.3	エンコード例	35
5 章	PDUの構成及びPDUの符号化	37
5.1	PDU構造	37
5.2	クラス-Base-1 (マルチキャスト通信) PDU	39
5.3	クラス-Base-2 (生存信号) PDU	40
5.4	クラス-Opt-2-a (障害情報)	42
5.5	クラス-Opt-3 (1 対 1 通信) PDU	44
6 章	適合性	45
6.1	適合性要求事項	45
付属書 A	システム用TCD	46
付属書 B	ベンダー名コード一覧	47
付属書 C	実装上の注意事項	48
C.1	メッセージ長	48
C.2	TCPコネクション管理	48
C.2.1	コネクションの補足制御情報	48
C.2.2	1 対 1 通信経路異常の検知	48
C.2.3	その他注意事項	49
C.3	メッセージの優先制御	49
C.4	TCDアクセス権制御	50
C.5	ノード内状態変化の統計情報とログ	50

C.6 拡張性	51
付属書D . (参考) 処理手順	52
D.1 マルチキャスト通信の送受信処理	52
D.1.1 送信処理	52
D.1.2 受信処理	52
D.2 1対1通信の送受信処理	53
D.2.1 送信処理	53
D.2.2 受信処理	54
D.3 通番と通番バージョン管理処理	55
D.3.1 マルチキャスト通信での処理	55
D.3.2 1対1通信での処理	58
D.4 メッセージの分割・組み立て	60
D.4.1 メッセージの分割・組み立て処理	60
D.4.2 分割、再組み立て用ヘッダ情報	61
D.4.3 メッセージ分割アルゴリズム	62
D.4.4 メッセージの組み立てアルゴリズム	63
D.5 ノード生死状態監視処理	67
D.6 障害情報送信方式	68
付属書E . (参考) 実装例	69
E.1 バイトオーダーの問題	69
E.2 アラインの問題	71
E.3 生存信号	72
E.4 障害情報の作成方法	74
付属書F . (参考) 応用例	75
F.1 ノード状態の表示	75
F.2 障害情報表示	76
F.3 システム監視ツール用エラーメッセージ仕様	77
付属書G . (参考) 自律分散システムの狙い	78
G.1 自律分散システムの狙い	78
G.2 自律分散プロトコルの特長	79
付属書H . 連想配列の狙い	80
H.1 特徴	80
H.2 本実装案での考え方	81

まえがき

この仕様は、（財）製造科学技術センター F A オープン推進協議会 分散型製造システム専門委員会（以下、「提供者」という）にて策定した自律分散インタフェースの機能及びプロトコルの仕様です。

なお、本仕様には、第三者より提供者に対して無償の通常実施権を許諾された特許が含まれています。本仕様書の利用者は、本仕様書の規定に従う対象製品を製造、販売するためのみ、当該特許に係る無償の通常実施権が許諾されます。但し、本仕様書の利用者が、無償の通常実施権を許諾している第三者に対し、対象製品に係わり特許に基づく権利主張をした場合、当該の利用者に対する関係では、本通常実施権は、初めから無かったものとします。

本仕様の内容に関する技術支援については、提供者がその責任において行うものとします。但し、提供者より紹介された第三者が、別途有償で行う場合があります。

また、本仕様の利用者は、以下を遵守ください。

- 本仕様書の利用者は、本仕様書記載内容に関し、改変を行うことは出来ません。但し、改変に関して提供者が承諾した場合は、この限りではありません。
- 本仕様書の利用者は、本仕様書記載の機能仕様に従い、かつEthernet上で使用することを条件として、ハードウェアまたはソフトウェア（以下、「対象製品」という）を製造、作成、使用又は販売するために、本仕様を無償で利用することができます。
- 本仕様を利用して、作成した成果物を稼働させる責任は、全て作成者が負うものとし、本仕様の提供者は一切の責任を負わないものとします。

0章 序文

従来の通信プロトコル標準は相手を特定するいわゆる1対1通信を基本としているものが多い。1対1通信は送達確認が容易で、信頼性の高い通信手段を提供する上では利点が多いが、通信の送受信者以外の第三者からの監視、受信者がダウンしたときのバックアップ、システム変更に伴う通信相手の切り替え等には多くの困難を伴う。特に工場自動化システムのように分散システムの場合に、これらは大きな問題となる。

自律分散システムは放送型の通信を基礎とし、各ノードが自律的に情報を選択して動作することにより全体が機能するシステムである。相手を特定しない放送型の特徴として、前記のような問題は生じない。

自律分散プロトコルはこのようなシステムにおいて使われるべき通信標準として定められた。

オープンな通信ネットワークとして広く普及しているTCP,UDP/IP および、Ethernet をそのまま使用し、その上に自律分散システムに必要な事柄を規定している。

1章 適用範囲

この仕様は自律分散の概念に基づいた通信プロトコル（自律分散プロトコルと称する）について次のことを規定する。

- (1) 下位層がTCP、UDP/IP、およびイーサネットであることを前提とした自律分散プロトコルについて規定する。
- (2) プロトコルの機能を規定する。
- (3) プロトコルデータ単位の符号化を規定する。
- (4) この仕様に適合する実装のための機能要求条件を規定する。

2章 プロトコルの前提

2.1 前提となる階層

本仕様で規定する自律分散プロトコルの機能を実装するために必要な階層を図1に示す。

アプリケーションプログラム
本仕様で規定する自律分散プロトコル
TCP/IP, UDP/IP
イーサネット (IEEE802.3 準拠)

図1 前提となる階層

2.2 前提となる通信プロトコル

自律分散プロトコルを搭載するためには、各機器に表1に示すRFCに準拠しているインターネットプロトコル(IP)を搭載していることを前提とする。

表1 前提となるRFC一覧

階層	プロトコル	準拠対象 (RFC番号)	準拠 レベル	備考
物理層		IEEE802.3準拠	必須	
データリンク層	イーサネットフレーム	RFC 894	必須	
	ARP	RFC 826	必須	
ネットワーク層	IP	RFC 791	必須	
	ICMP	RFC 792	任意	
	ブロードキャスト関係	RFC 919 ,RFC 922	必須	
	サブネット関係	RFC 950	勧告	未サポート機器はシステム制限を明確化要
トランスポート層	UDP	RFC 788	必須	
	TCP	RFC 793,761,675	選択	1対1通信サポート時必須

[注意]

サブネットワークをサポートしない機器は、機器の取り扱い説明書あるいはマニュアルに次の制限事項として次の2点を記載する。

-) 「当該製品はサブネットワークをサポートしていない」こと。
-) 「当該製品を使用するシステムでは、サブネットによるシステム構築はできない」こと。

3章 用語の定義

3.1 ADP (Autonomous Decentralized Protocol)

本仕様で規定する自律分散プロトコル。

3.2 機器

LANに接続されるシステムコンポーネント、たとえば物理的なコントローラ、計算機、制御機器など。

3.3 LANセグメント

あるイーサネットフレームが流れる範囲のLANで、リピータ、ブリッジなどで接続されるLAN。IPアドレスのネットワークアドレスが同じ範囲のLAN。

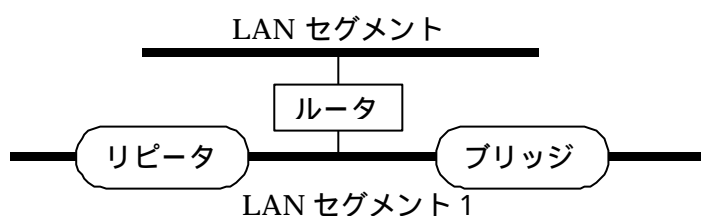


図2 LANセグメント

3.4 論理ノード

データフィールドに所属する機器。データフィールド内で一意に論理ノードを識別するための番号が論理ノード番号 (LNN)。以降、略してノード番号と呼ぶ。ノード番号は、データフィールド内で一意になるように割り当て、1～4095の範囲で指定する。論理ノード番号0は、自ノード内通信用である。

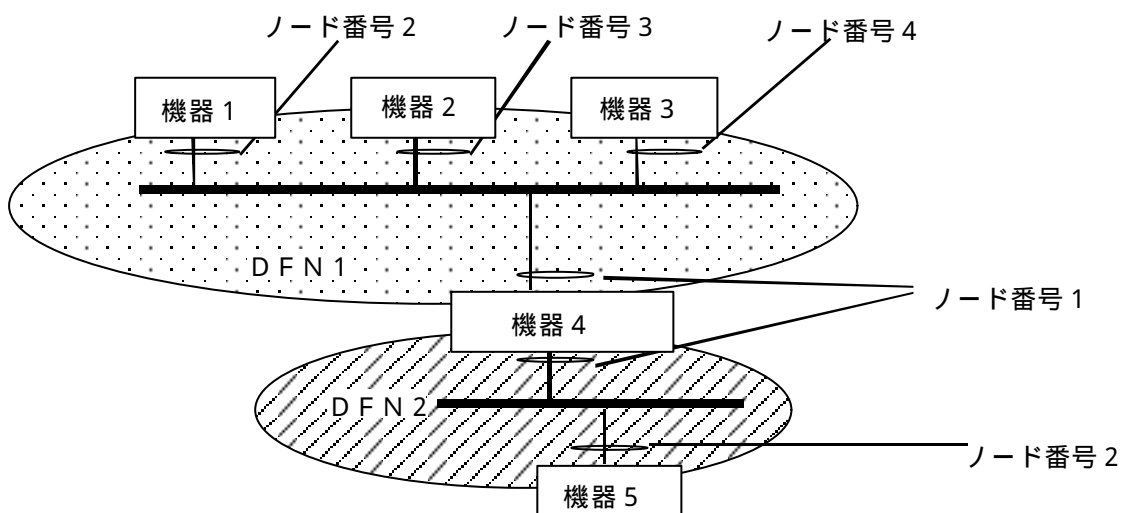


図3 論理ノードと論理ノード番号

3.5 データフィールド (DF)

特定の性格を持つ情報によって構成される自律分散データ(ADPメッセージ)が流れる場。

データフィールドに属するノード間で自律分散プロトコルによる通信が行える。したがってシステム内の機器は、必ずひとつ以上のデータフィールドに属さなければならない。一つのデータフィールドは、IPアドレスのネットワークアドレスあるいはサブネットワークアドレス単位で設定する。データフィールドを分散システム内で一意に識別するための番号がデータフィールド番号 (DFN)。データフィールド番号は、システム内で一意となるように割り当て、1~255の範囲で設定する。なおデータフィールド0は、自ノード内通信用に予約されている。

[注意事項]

- データフィールドは、LANセグメントをまたいで設定できない。LANセグメント毎にデータフィールド番号を割り当てる。
- ひとつのLANセグメントに複数のデータフィールドを定義できる。

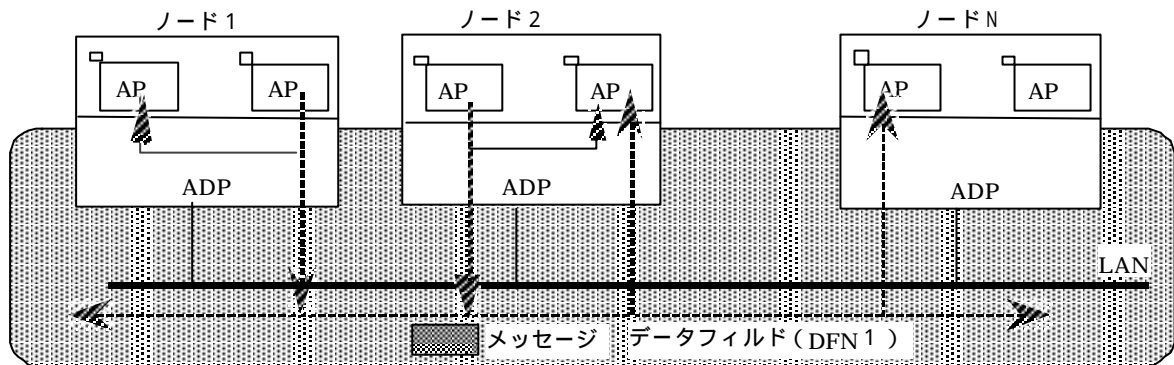


図4 データフィールドとデータフィールド番号

3.5.1 ローカルデータフィールド

自ノードが直接接続するデータフィールド。

3.5.2 リモートデータフィールド

ルータ、ゲートウェイを介して接続され自ノードが間接的に接続されるデータフィールド。

[例] データフィールド1とデータフィールド2に接続しているノード1から見たローカルデータフィールドは、データフィールド1と2で、とリモートデータフィールドはデータフィールド3と4である。

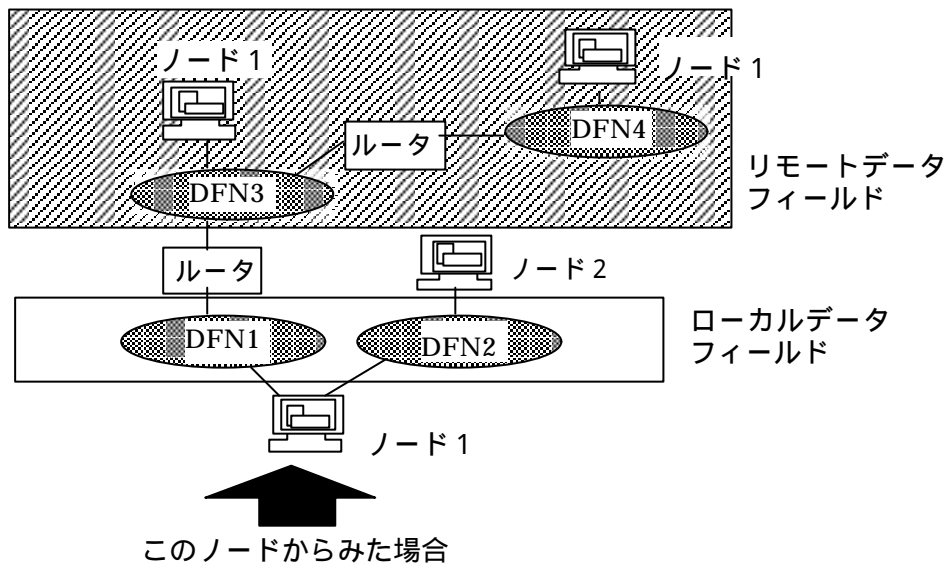


図5 ローカルデータフィールドとリモートデータフィールド

3.6 ドメイン

複数のデータフィールドをグループ化したデータフィールドの上位概念。通常一つのドメインはローカルな拠点として定義され、他ドメインとは、広域通信網を介して接続する。

システム内でドメインを一意に識別するための番号がドメイン番号 (DMN)。ドメイン番号は、システムで一意となるように割り当て、1～64の範囲で指定する。また、このドメイン番号を0にすると(省略すると)、システムは同ドメインを指定したとみなす。現在ドメイン番号 (DMN) は0固定で使用する。

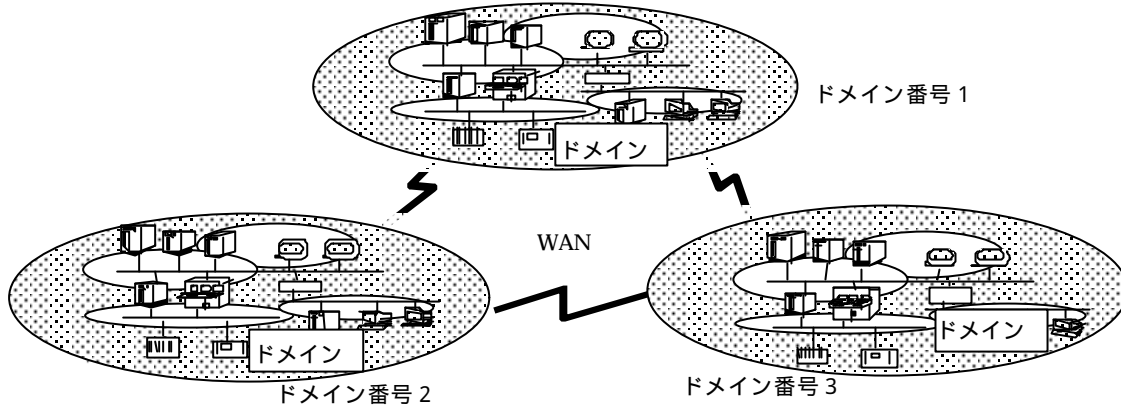


図6 ドメインとドメイン番号

3.7 マルチキャストグループ

データフィールド内に設けるノードの集合。マルチキャスト送信は、このマルチキャストグループを指定して行う。また、マルチキャスト受信したいノードは、マルチキャストグループに加入する必要がある。一つのデータフィールド内に複数のマルチキャストグループを設けることができ、各ノードは自分が属するデータフィールド内の複数のマルチキャストグループに加入することができる。データフィールド内でマルチキャストグループを識別するための番号がマルチキャストグループ番号 (MGN) である。マルチキャストグループ番号は、データフィールド内で一意となるように割り当て1～255の範囲で設定する。マルチキャストグループ0は、システム用で生存信号の送受信で使用する。

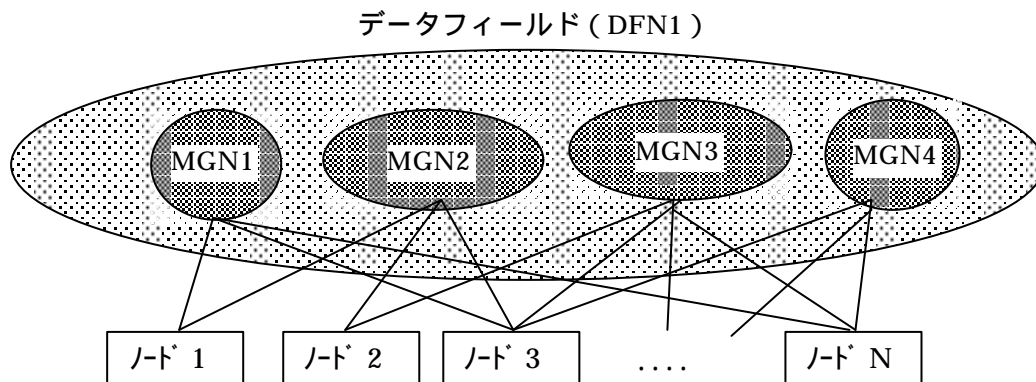


図7 マルチキャストグループとマルチキャストグループ番号

3.8 ブロードキャスト

全ノード宛の送信。ブロードキャスト送信されたメッセージは全てのノードで受信を行う。

3.9 マルチキャスト通信

自律分散プロトコルで規定する通信方式の一つで、UDPプロトコルのブロードキャストをベースとした通信。1回の送信要求で送信要求したマルチキャストグループに属す複数のノードへ同一メッセージを送信する。

3.10 1対1通信

自律分散プロトコルで規定する通信方式の一つで、TCPをベースとした通信。通信に先立ち各ノード間に1つ1対1通信経路(TCPコネクション)を確立し、APからの送信要求時、送信メッセージを指定相手ノードへ送信する。

3.11 生存信号

あるデータフィールドに属するノードが、自分(自ノード)が生存していることを、そのデータフィールドに属する全てのノードに知らせるために、周期的にそのデータフィールドにブロードキャストするメッセージ。

3.12 障害情報

あるデータフィールドに属するあるノードが、そのノードにおいて発生した障害についての情報を、そのデータフィールドに属する全てのノードに知らせるために、周期的にそのデータフィールドにブロードキャストするメッセージ。

3.13 AP(Application Program)

ADPの上位レイヤに位置する業務処理のプログラムで、ユーザのプロセスやタスクのこと。

3.14 PDU(Protocol Data Unit)

下位レイヤからADPに受け渡されるデータあるいは下位レイヤにADPが受け渡すデータの単位。ADPヘッダ部とデータ部とから成る。

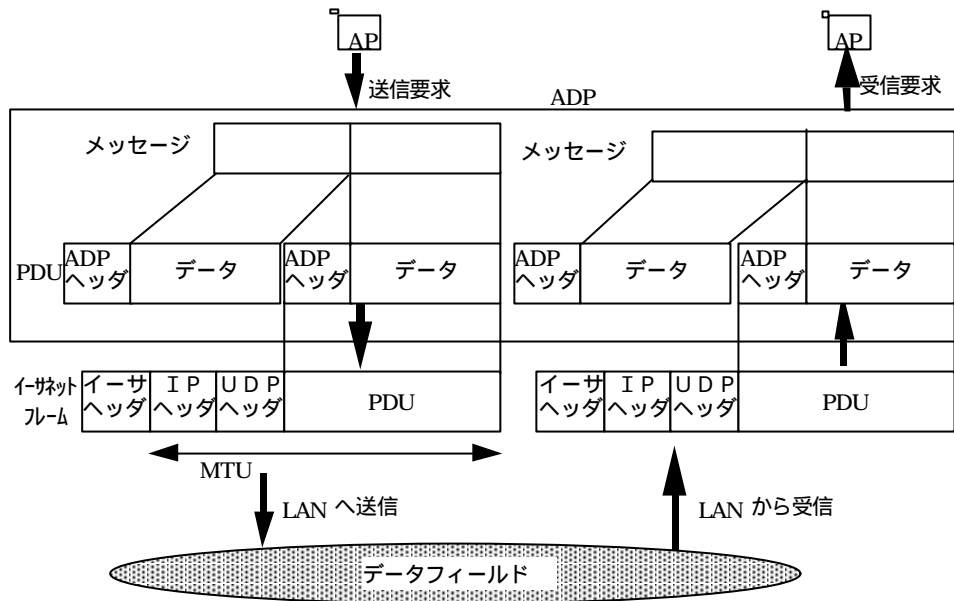


図8 メッセージとPDU

3.15 メッセージ

上位レイヤに位置するAPからADPに受け渡されるデータあるいは上位レイヤのAPにADPが受け渡すデータの基本単位。

3.16 トランザクション

AP内やシステムでの処理。トランザクションを識別するためのコードがトランザクションコード (TCD)。APはTCDを指定してメッセージを送信、あるいは自ノードに届いたメッセージの中からAPの処理 (トランザクション) に必要なものを受信する。ADPはTCDをADPヘッダに格納し通信を行う。TCDは、各データフィールド毎に一意に定義する。トランザクションをユーザトランザクションとシステムトランザクションの2種に分けている。それぞれにユーザ用TCD、システム用TCDを次の範囲で割り当てている。システム用TCDについては、付属書Aを参照のこと。

ユーザ用TCD	:	1 ~ 59999
システム用TCD	:	60000 ~ 65534

3.16.1 ユーザトランザクション

業務処理であるAP内のトランザクション。ユーザトランザクション識別のためのコード (ユーザ用TCD) はユーザが業務処理毎に割り付け使用する。

3.16.2 システムトランザクション

ADPが各種イベント（異常状態、内部状態変化など）を検知した時に生成し、APに通知するために使用するトランザクション。システムトランザクション識別のためのコード（システム用TCD）は、ADPの各種プロトコル機能実装者が使用する。

APはシステム用TCDが付与されたメッセージを、ノードモード（オンラインモード、テストモード）や、入出力制御に関係なく送受信しなければならない。システムトランザクションとシステム用TCDの割り付けを付属書Aに示す。

本仕様では、TCD = 6 0 0 0 3（生存信号）のみを定義している。

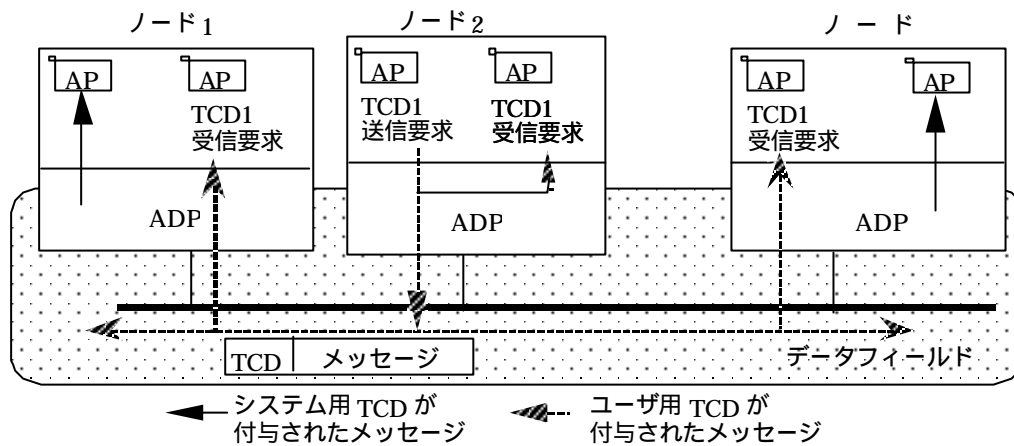


図9 トランザクションコード

3.17 実装者

自律分散プロトコルを機器に実装する企業・人。

3.18 ユーザ

自律分散プロトコル搭載機器を組み合わせてシステムを構築し使用するエンドユーザ。

4章 プロトコルの機能

4.1 クラス-Base-1 (マルチキャスト通信)

4.1.1 機能

送信側は、TCDの付与されたメッセージを指定したデータフィールド内のマルチキャストグループにマルチキャスト送信する。指定されたマルチキャストグループ内の各ノードは、自律的に、必要なTCDのメッセージのみを取り込む。送信時、相手ノードのアドレスを指定しないため拡張性のある通信が可能となる。

このマルチキャスト送信したメッセージを受信するためには、各ノードはあらかじめマルチキャストグループへ加入しなければならない。データフィールド内の全てのノードにメッセージを送信したい場合は、全ノードが同じマルチキャストグループに加入する。

[参考]

<マルチキャスト通信の特徴>

自律分散システムのベースとなる通信方式

複数ノードへ同時に同一メッセージを送信でき効率が良い(1対N型通信)

リアルタイムデータの通信向き

定常的に発生するデータを複数のノードへ一度に送達する場合に有効

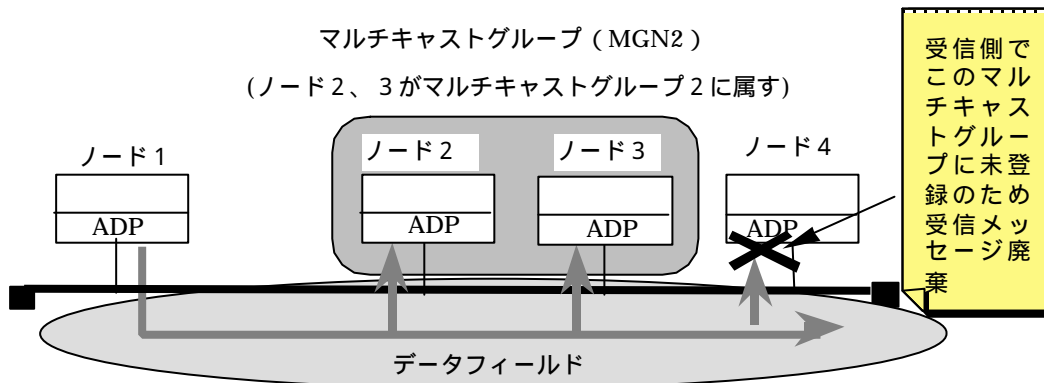


図10 マルチキャスト通信

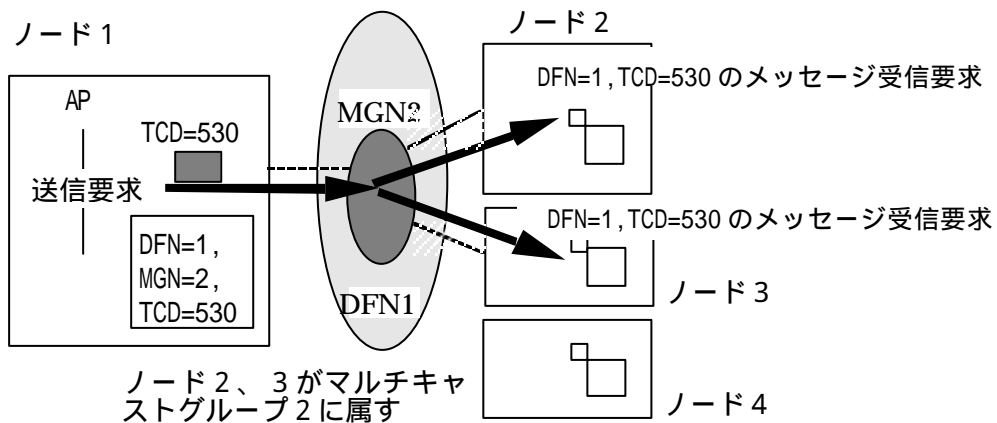


図11 マルチキャスト通信の送受信

マルチキャスト通信は、UDPプロトコルの上位プロトコルである。マルチキャスト送信したデータは、LAN上へブロードキャストされ各ノードに受信される。そしてADPレベルで受信メッセージの宛先マルチキャストグループ番号に自ノードが登録していない場合、そのメッセージを廃棄する。登録されている場合にADPはAPへこのメッセージを渡す。

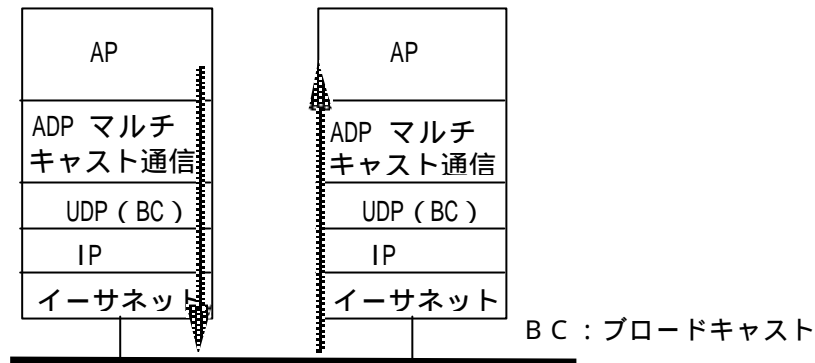


図12 UDPベースの通信

4.1.2 マルチキャストグループの管理方法

マルチキャストグループは、データフィールド内で定義されるノードの集合である。各ノードは自由にマルチキャストグループに加入、あるいは離脱することができる。

4.1.2.1. マルチキャストグループ番号とUDPポート番号の割り当て方

マルチキャストグループ番号の設定可能範囲は、1～255である、なお、マルチキャスト番号0は生存報告用を使用するためにシステムで予約されている。

各マルチキャストグループが受信するメッセージモード（「4.5 テスト支援機能」参照）毎にUDPポートを割り当てる必要がある。各ノードが使用するマルチキャストグループ番号のメッセージモードとUDPポート番号の対応付けは、データフィールド内で統一されていなければならない、データフィールド毎に管理しなければならない。

4.1.2.2. マルチキャストグループへの加入と離脱

4.1.2.2.1. 加入

各ノードは、立ち上げ時などのタイミングでマルチキャストグループに割り当てられているUDPポート番号と結びつける。この操作にてマルチキャストグループへ加入が行なわれる。

[例] ソケットを使用する機器での加入例

- (a) マルチキャストMGNi用ソケットをインターネットプロトコル、データグラムタイプで生成。
- (b) ソケットにワイルドカードアドレス (IPアドレス0)、MGNiに割り当てられているUDPポート番号でバインド (bind) する。
- (c) 受信ソケットのバッファサイズを受信量に応じて調整する。

4.1.2.2.2. 離脱

マルチキャストグループへの加入で結び付けたノードと、UDPポート番号との対応を解除する。解除することにより、マルチキャストグループから離脱する。

[例] ソケットを使用する機器での離脱例

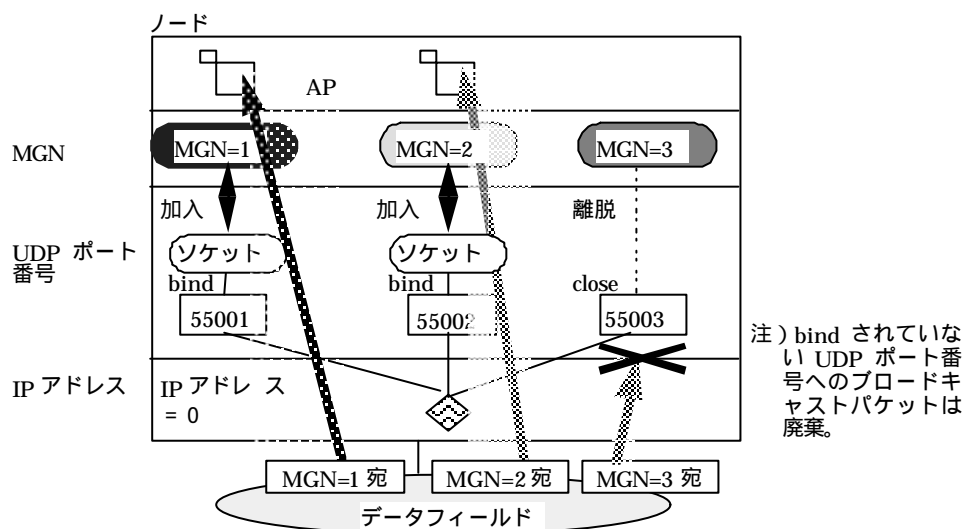


図13 マルチキャストグループへの加入、離脱

4.1.2.2.3. マルチキャスト送信ポートの作成

マルチキャストグループへ送信するための送信元UDPポート番号をデータフィールド単位に割り付ける。

[参考]

送信元UDPポート番号は、1つのポートにて複数のマルチキャストグループへの送信を共用し、また、データフィールド毎に異なったポート番号でも設定可能とすることを推奨する。なお、データフィールド内の各論理ノードでは、同一送信元UDPポート番号を使用することを推奨する。

[例] ソケットを使用する機器での作成例

- (a) マルチキャスト送信用ソケットをインターネットプロトコル、データグラムタイプで生成する。
- (b) ソケットにワイルドカードアドレス (IPアドレス0)、送信用UDPポート番号でバインド(bind)する。
- (c) ソケットにブロードキャスト使用を宣言 (setsockopt) する。

4.2 クラス-Base-2 (生存信号送信)

4.2.1 機能

データフィールドに属するノードが、自ノードが生存していることをそのデータフィールドに属する全てのノードに通知するために送信する。これにより、データフィールド上のノードの構成状態を互いのノードで認識できるようになる。あるデータフィールドに属する全てのノードは、生存信号を送出可能な状態にあるかぎり、生存信号を周期的にそのデータフィールドに送出し続ける必要がある。

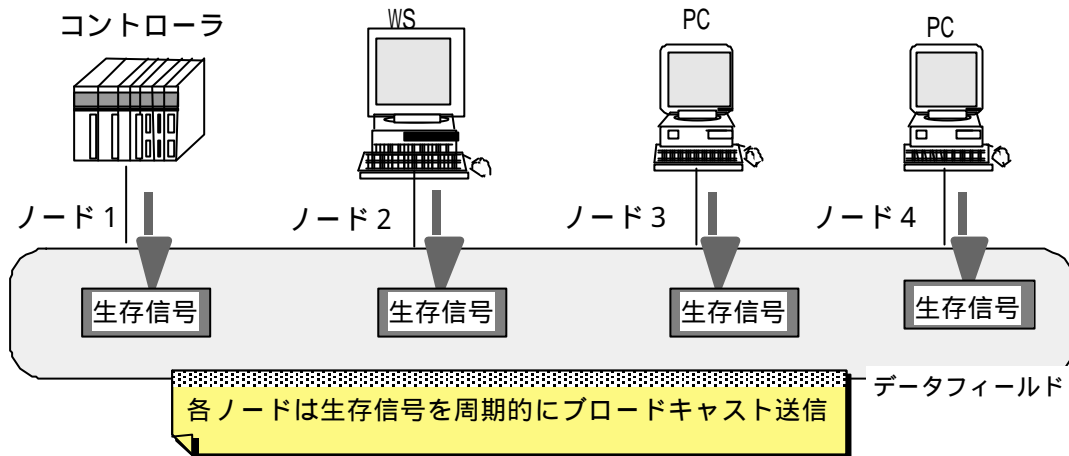


図14 生存信号の送信

[参考]

生存信号は、ノードの構成管理に使用する。送出された生存信号は、そのデータフィールドに属する任意のノードで受信され、そのデータフィールドに属する任意のノードの生存を認識することができる。あるデータフィールドに属するノードが、そのデータフィールドに属する他のノードの状態（生/死：正常か/異常か）を識別し、システム全体の構成を管理および、制御することができる。

[例]

ノード1の業務バックアップを行っているノード2があった場合に、ノード2では常にノード1の生存信号を監視し、ノード1が「死」と判断された場合には、ノード1の業務をノード2で引き継ぐということが可能になる。また、ノード1の「生」を認識することにより、ノード2にて行っていたノード1の業務を停止させることも可能となる。

4.2.2 生存信号送信方式

4.2.2.1. 送信先データフィールドの範囲

各ノードは、ノードが直接接続されているネットワークに対応するデータフィールドに対してのみ生存信号メッセージを送信する。生存信号とそれに基づくノード生死監視は、ローカルデータフィールド、すなわち各データフィールドごとにクローズして行うものであって、ルータ等を介して間接的に接続されるデータフィールド（リモートデータフィールド）に対して生存信号メッセージの送信は行わない。なお、複数のデータフィールドに接続されているノードの場合、各データフィールドごとに生存信号を送信する。

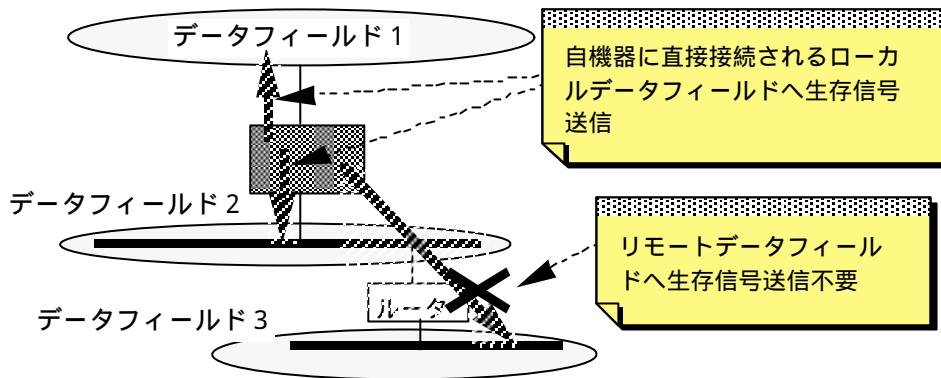


図15 生存信号送信先データフィールド

4.2.2.2. 生存信号送信間隔

生存信号メッセージの送信間隔は、論理ノード毎に設定する。複数のデータフィールドに属している場合はデータフィールド毎に設定する。機器がオンライン状態になった後、この決められた送信間隔で生存信号メッセージを送信する。

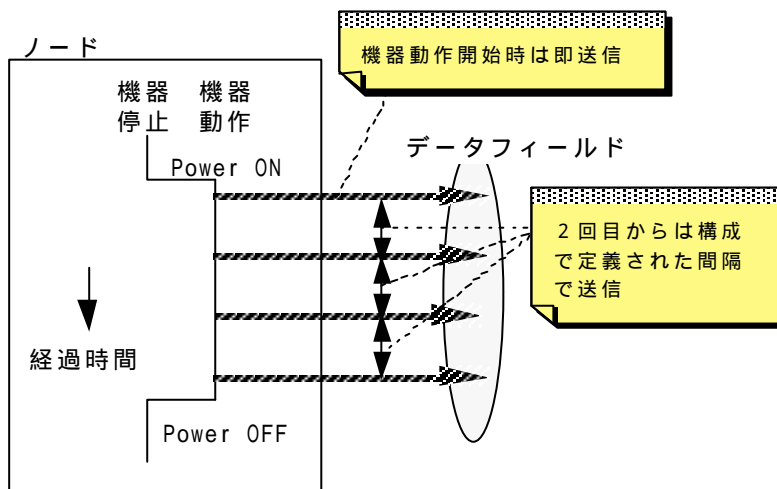


図16 生存信号の送信間隔

4.2.3 生存信号によるノードの生死判定

生存信号を受信することにより、任意のノードの生死を知るために使用する。生存信号を受信するノードは、以下の方法にて任意のノードの「生」と「死」を判断する。

- あるデータフィールドにおいて、あるノードからの生存信号を一度も受信できなければ、そのノードを「死」と判断する。
- あるデータフィールドにおいて、あるノードからの生存信号を一度受信できれば、その時点でそのノードを「生」と判断する。
- あるデータフィールドにおいて、あるノードからの生存信号を受信し、その時点でそのノードを「生」と判断できても、その後、受信した生存信号の生存信号ヘッダに格納されている生存信号タイムアウト監視時間 (al_tm_out) の間、新たに生存信号が受信できなければ、その時点で、そのノードを「死」と判断する。処理の詳細は「付属書 D.5ノード生死状態監視処理」を参照。

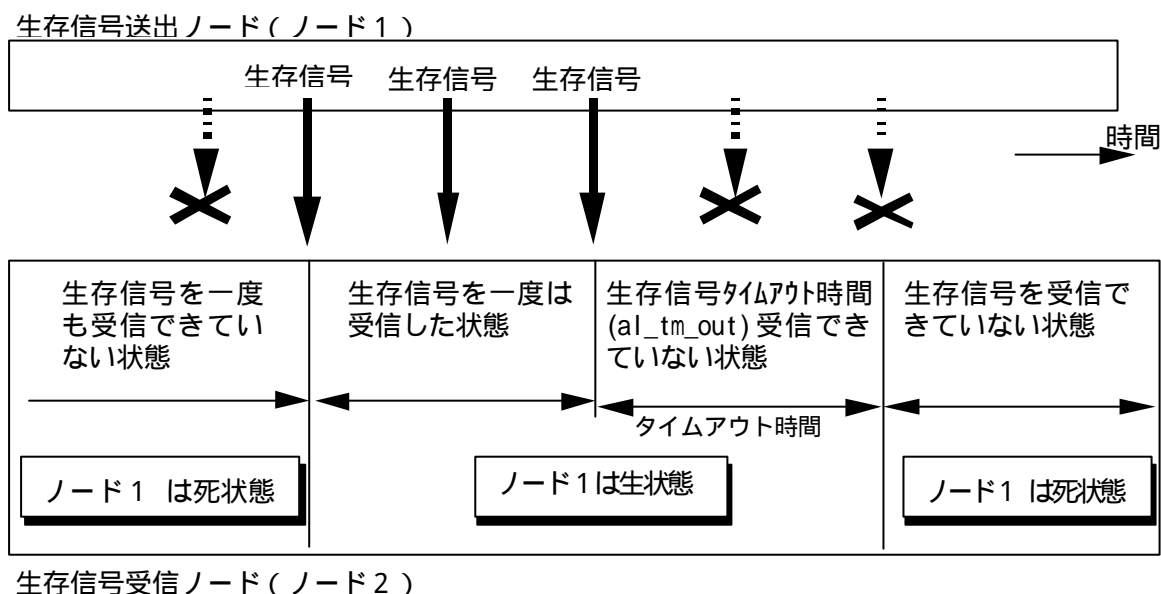


図17 生存信号によるノード状態の監視シーケンス

4.2.4 UDPポート番号の割り当て方法

生存信号メッセージは、自機器のローカルデータフィールドに対してマルチキャストで送信する。生存信号は、マルチキャストグループ0を使用し、このグループはデータフィールド内の全ノードが無条件に受信するブロードキャスト用として位置づけている。よって、生存信号の送信先マルチキャストグループ番号は0を設定する。

各機器は、生存信号メッセージの送信先UDPポート番号をユーザが任意に選んだ番号で設定でき、送信できるようにしておく必要がある。生存信号用のUDPポート番号は、各データフィールド毎に任意の番号をユーザが選択し構築する。

[注意]

同一データフィールド内に接続されるノードの生存信号用UDPポート番号は、同一の番号を使用する（図18 参照）。

また、自ノードが生存信号メッセージを受信する場合には、生存信号メッセージを受信するためポート番号のUDP受信ポートを生成する必要がある。受信UDPポート番号は、そのデータフィールドで取り決めたUDPポート番号を使用する（図19 参照）。

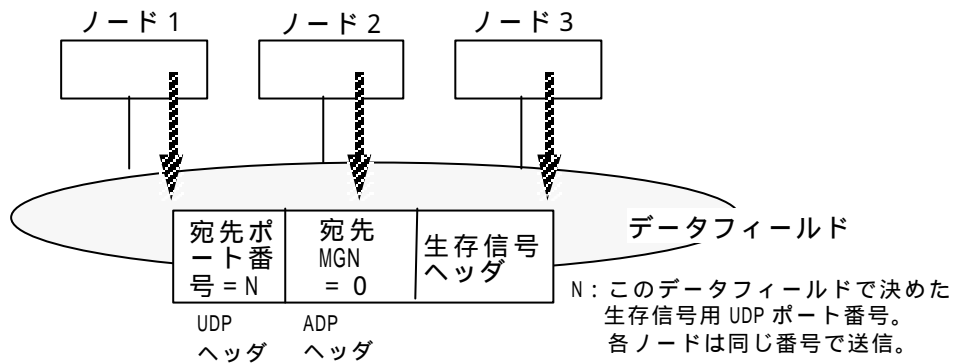


図18 生存信号の宛先UDPポート番号

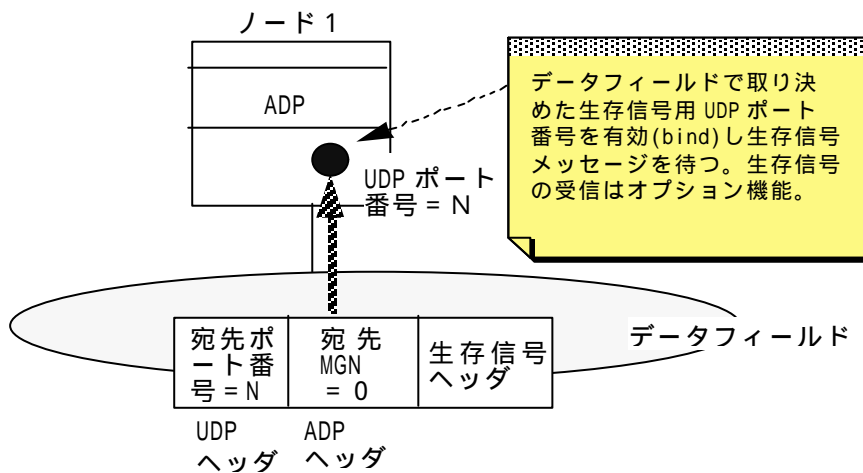


図19 生存信号の受信用UDPポート番号

4.3 クラス-Opt-2-a (障害情報送出)

4.3.1 機能

生存信号の拡張情報として障害情報を送出する。あるデータフィールドに属するノードが、そのノードにおいて発生した障害についての情報を、そのデータフィールドに属する全てのノードに知らせるために使用する。

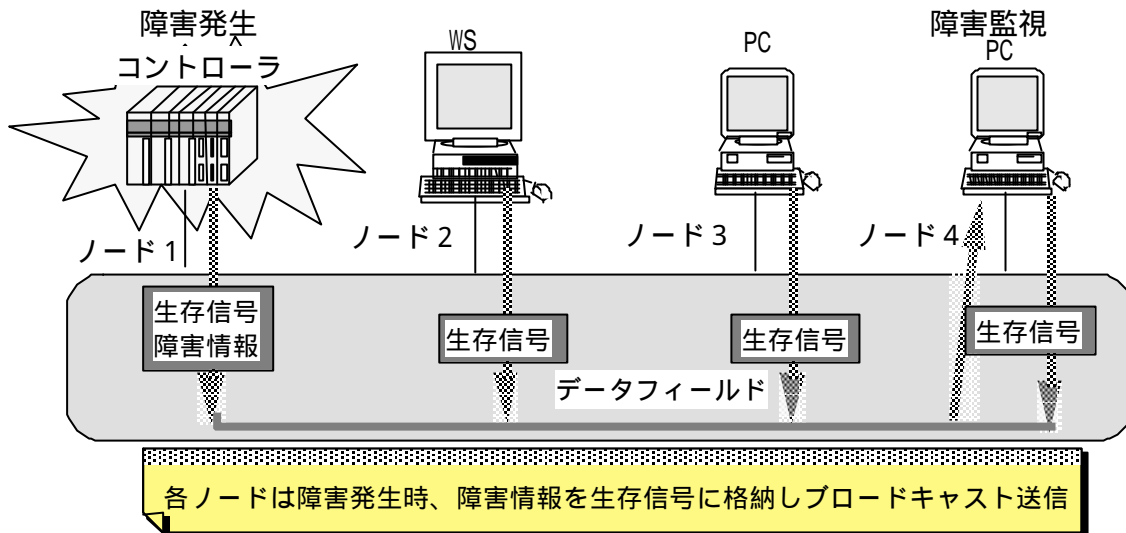


図20 障害情報の送信

4.3.2 障害情報

障害情報には、以下が含まれる。障害発生時、各ノードで障害内容を下記情報に変換する。

◆ ノード内モジュール生死情報

モジュール生存情報は、送出時点のモジュールの状態を通知する。

[参考]

モジュールとは、ソフトウェアモジュールであっても、ハードウェアモジュールであってもかまわない。モジュールの定義は、生死情報を送出する対象となるもの（結果として生死監視の対象となるもの）を、そのノードにおけるモジュールと定義する。通常、ソフトウェアモジュールとしては、プロセス/タスクがモジュールに対応する。

◆ 障害番号（エラー番号）

ノード内で発生した障害の内容を番号で通知する。障害番号は、前回送出以降に発生した障害についてのみ報告する。障害が回復されず、障害状態となったままの場合には、継続してその障害番号を送出する。発生した障害について、その番号（複数可能）を送出することにより、監視APにて、障害の内容を表示することができるようになる（付属書 F.2 参照）。また、障害情報を受信しているノードでは、障害の発生したノードでの障害の内容を知ることができる。

◆ 機器固有のオプション情報

機器固有の障害情報を機器固有のフォーマットで通知する。

[例]

コントローラ内のステータスレジスタの内容をそのまま送出的ることにより、後述の監視APにて、その内容を表示させることができる。(付属書 F.2 参照)

4.4 クラス-Opt-3 (1対1通信)

4.4.1 機能

1対1通信は、TCPをベースとした通信です。通信に先立ち各ノード間に1つ1対1通信経路 (TCPコネクション) を確立し、APからの送信要求時、送信メッセージを指定相手ノードへ転送する。TCPコネクションを使用するために、構築時にあらかじめ相手ノード毎にIPアドレスやコネクション接続用ポート番号を定義する必要がある。

この通信方式は、1対1で確実に相手ノードへ送達する、あるいは1対1の大量データ転送が要求されるデータの通信に有効である。

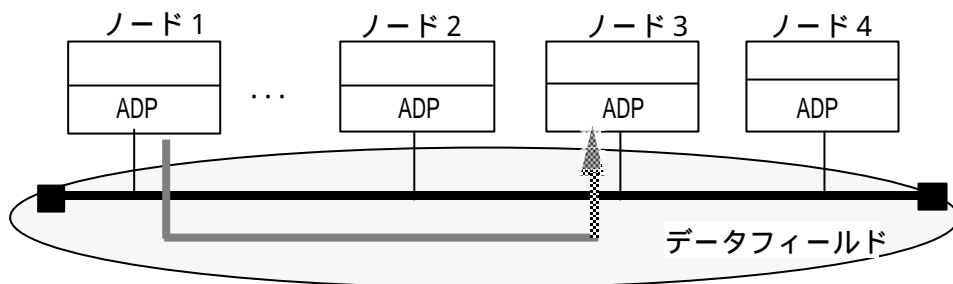


図21 1対1通信

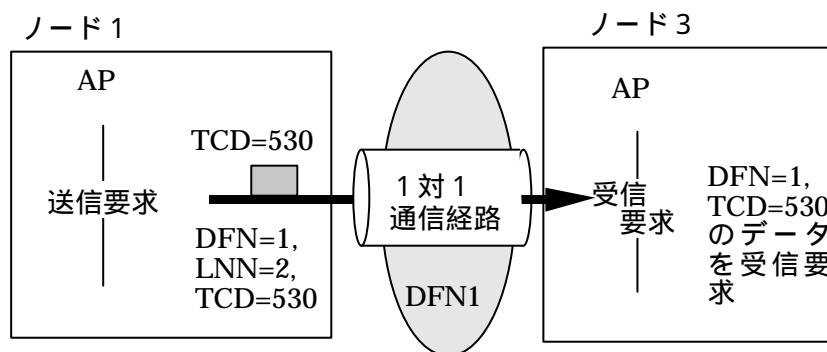


図22 1対1通信経路と送受信

1対1通信は、TCPプロトコルの上位プロトコルである。1対1通信の送信メッセージは、ノード間のTCPプロトコルによりTCPコネクション経由で相手ノードへ転送される。

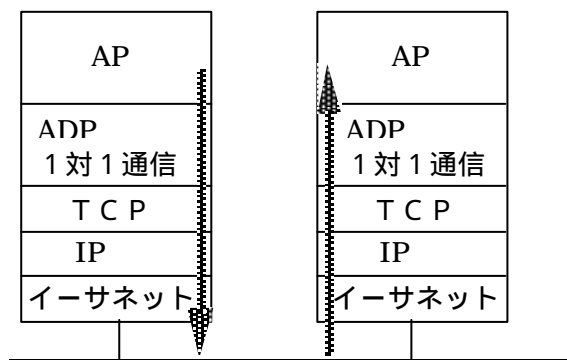


図23 TCPベースの通信

4.4.2 1対1通信用TCPコネクション管理

4.4.2.1 コネクションとTCPポート番号の割り当て方

1対1通信用に、1つのノード内でユニークなTCPポートを割り付ける。1対1通信用コネクションは、ふたつのノード間であらかじめ取り決めしたTCPポート番号を使用しコネクションを確立する。

[例]

ノード1は、ノード2、3と1対1通信用のコネクションを確立します。この際、コネクションを確立するための情報として相手ノード番号に対するIPアドレスと相手TCPポート番号が必要である。

ノード1とノード2間のコネクション情報は、IPアドレスIP1とIP2およびTCPポート番号10000と20000である。ノード1とノード2それぞれこの4つの情報を定義しコネクション確立処理を行う。

コネクション確立は、あらかじめコネクション確立を要求する側とその要求を待つ側とを決める。コネクション確立を要求する側は、コネクション確立が相手ノード停止などで失敗しても定期的のリトライできるようにする。コネクションが確立されると1対1通信ができる。

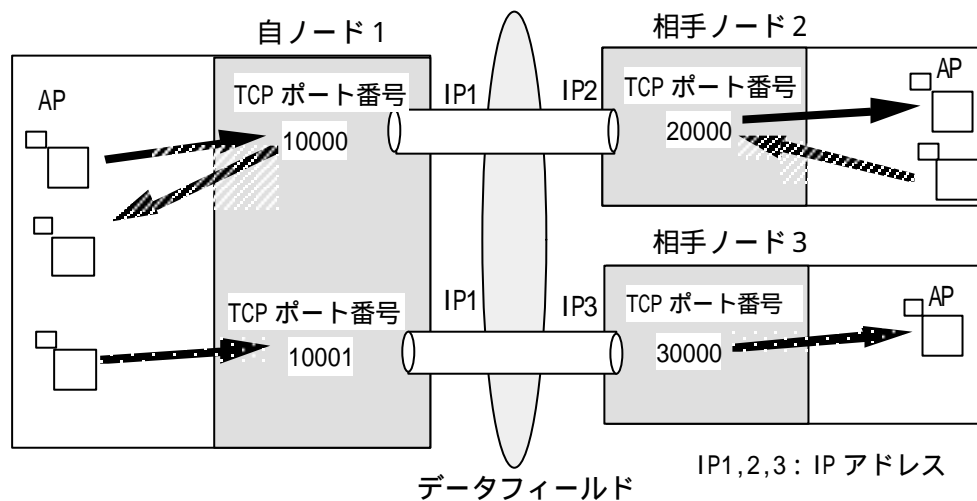


図24 TCPコネクションとTCPポート番号の割り当て

4.4.2.2 1対1通信コネクション状態

自律分散プロトコルでは、1対1通信のコネクションの状態を次の3つの状態で管理する。

◆ ホールト状態

通信相手ノードがシステムに登録され、かつ、そのノードに関するシステム管理テーブルが初期化された状態である。この状態では、メッセージ通信はできない。他ノードとのコネクションを確立することが可能な状態である。TCPの状態が、CLOSED状態である。

◆ オープン状態

この状態は、コネクション確立処理中の過渡的な状態で、コネクションが確立されるとリンク状態に自動的に遷移する。この状態では、メッセージ通信はできない。コネクション確立処理中とは、通信相手ノードに対してコネクション確立要求し相手からの応答を待っている状態（TCPの状態がSYN_SENT状態）、あるいは相手ノードからのコネクション確立要求を待っている状態（TCPの状態がLISTEN状態あるいはSYN_RECEIVED状態）のいずれかである。通信相手ノードとのコネクション確立が完了した時点（TCPの状態がESTABLISHED状態）でリンク状態に遷移する。コネクション確立が失敗した場合は、オープン状態を保持する。また、オープン状態が確定している状態でリンク状態への遷移を指示された場合、一度ホールト状態に遷移した後、再度リンク状態へ遷移させる。この時一時的にホールト状態を経由する。

◆ リンク状態

通信相手ノードとの間にコネクションが確立でき、メッセージの送受信が可能な状態（TCPの状態がESTABLISHED状態）です。この状態でホールト状態への遷移指示が出た場合は、両ノードとも自動的にホールト状態に遷移します。TCPでは通常FINを発行し3ウェイハンドシェイク手順にてコネクションを解放しますが、本システムではRESETパケットを発行し瞬時にコネクションを切断する手順を採用しています。また、リンク状態でコネクション切断（RESET受信）、通信障害を検知するとホールト状態に自動的に遷移させます。

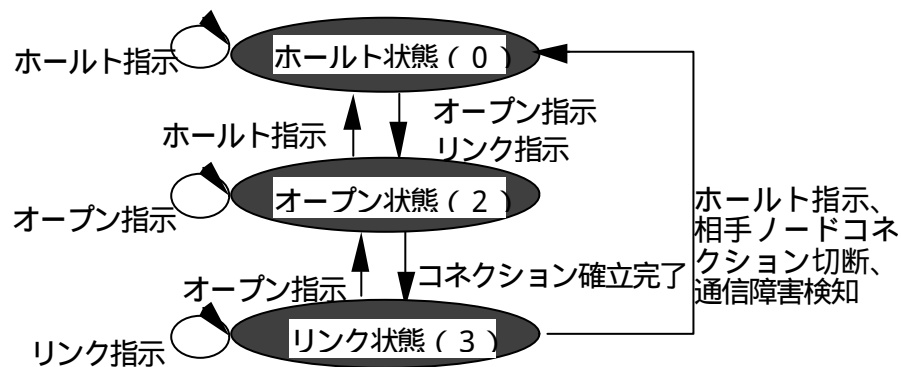


図25 TCPコネクションの状態遷移

4.5 テスト支援

4.5.1 機能

メッセージモードおよびノードモードにより、メッセージの送受信制御を行うことによりオンライン動作中のノードにテストメッセージが入り込むことなく、テストノード間のみでのテストが実施できる。テスト動作中のノードは、オンライン、テスト両方のメッセージを受信しシミュレーションテストを行うなどが可能となる。(図26 参照)

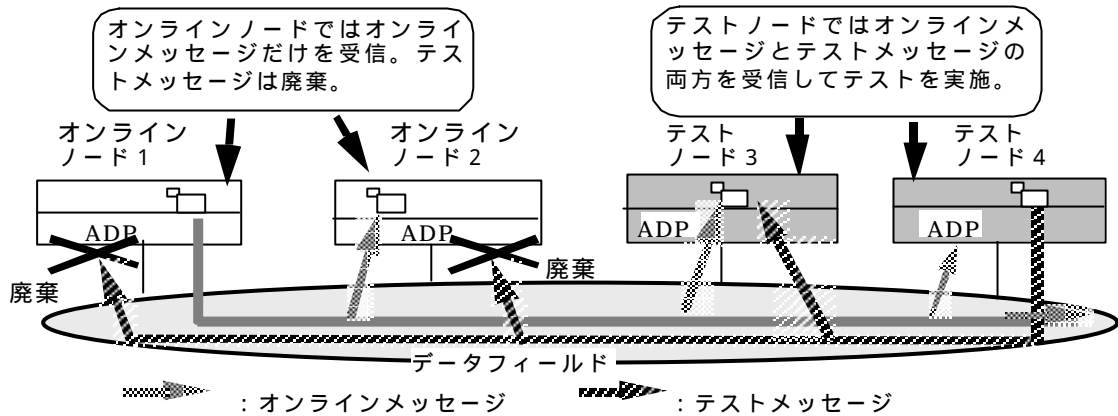


図26 テスト支援機能

4.5.2 メッセージモード

メッセージのモードとして次の2種類がある。

- (1) オンライン
- (2) テスト

4.5.3 ノードモード

ノードのモードとして次の2種類がある。

- (1) オンライン
- (2) テスト

4.5.4 メッセージ送受信制御

メッセージモードとノードモードにより、下表に示すメッセージの送受信制御を行う。

表2 メッセージ送受信処理の関係

ノードモード \ メッセージモード	オンライン		テスト	
	送信	受信	送信	受信
オンライン	可能	可能	対象外	可能
テスト	対象外	廃棄	可能	可能

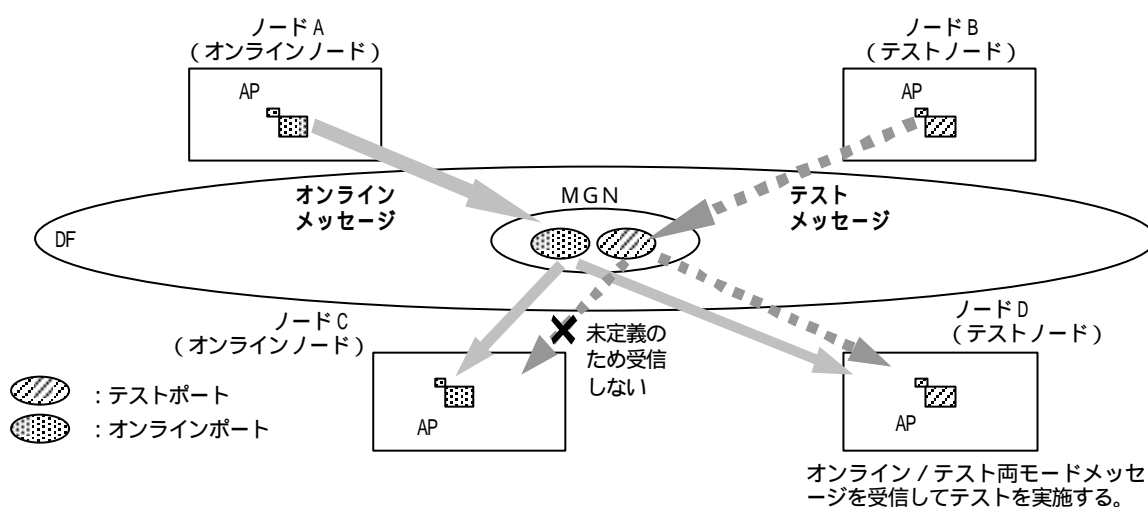
4.5.5 マルチキャスト受信ポートの割当

オンラインノードにてオンラインモードメッセージのみを受信する場合は、マルチキャストグループ1つに対して、1つのUDPポートを使用する。

テストノードにてオンラインモードメッセージとテストモードメッセージの両方を受信する場合は、1つのマルチキャストグループに対して、オンラインメッセージとテストメッセージで、別々のUDPポートを使用する。

[参考]

1つのマルチキャストグループでのオンラインメッセージ用のUDPポートとテストメッセージ用のUDPポートの対応付けは、データフィールド内で統一して管理したものを各ノードで使用しなければならない。



これは、オンラインポートとテストポートを共有しないことにより、同一データフィールド内にオンラインノードとテストノードが共存した場合でも、テストメッセージがオンラインノードに擾乱を生じさせないため、特にハードウェアレベルにて未定義ポートのメッセージを受信しない機器においては、CPUの負荷を低減する上で有効となる。

[例]

図28 に、1から255までのマルチキャストグループ番号とUDPポート番号（オンラインモード、テストモード）の対応付けの例を示す。

マルチキャストグループ番号	1	2	3	...	255	← 宛先MGN
UDPポート番号（オンライン）	55001	55002	55003	...	55255	} UDPプロトコル レベルの宛先 ポート番号
UDPポート番号（テスト）	57001	57002	57003	...	57255	

図28 MGNとUDPポート番号の割り当て

4.5.6 テスト支援機能処理

(1) 送信側処理

送信ノードで、送信するメッセージのモードをADPヘッダ内のメッセージモード (MODE) に格納し送信する。

(2) 受信側処理

受信ノードでは、自ノードのノードモードがオンラインノードである場合は、ADPヘッダ内のメッセージモード (MODE) を確認し、テストメッセージは廃棄し、オンラインメッセージだけを受信する。なお、このメッセージ廃棄は障害と見なさない為、廃棄に対し障害情報を送出する必要はない。自ノードのノードモードがテストノードである場合は、テストメッセージとオンラインメッセージのどちらも受信する。

4.6 メッセージ優先制御

4.6.1 機能

メッセージに優先レベルを割り付け、メッセージ送受信処理のスケジューリングを行う。優先度の低いデータに影響されことなく優先度の高いデータを優先的に送受信処理可能となる。

4.6.2 メッセージ優先レベル

メッセージの優先レベルは、最大7つあり、表3の規則で割り付ける。メッセージ優先制御処理を実装しないノードのメッセージ優先レベルは、0固定とする。

表3 優先レベル割付表

レベル	優先順位	割付用途	使用者
0	なし	優先制御対象外	ユーザ
1	高い	ユーザ用最高優先メッセージ	
2	↕	...	
...			
7	低い	ユーザ用最低優先メッセージ	

4.6.3 メッセージ優先制御処理

4.6.3.1 送信側処理

- ・送信ノードでは、送信するメッセージの優先度に応じメッセージに優先レベル1～7を割り付け、割り付けたメッセージ優先レベルの値をそのメッセージのPDUのADPヘッダ内のメッセージ優先レベル(PRI)に格納し送信する。
- ・送信ノードがメッセージ優先制御処理を実装しない場合は、ADPヘッダ内のメッセージ優先レベル(PRI)に0を格納し送信する。

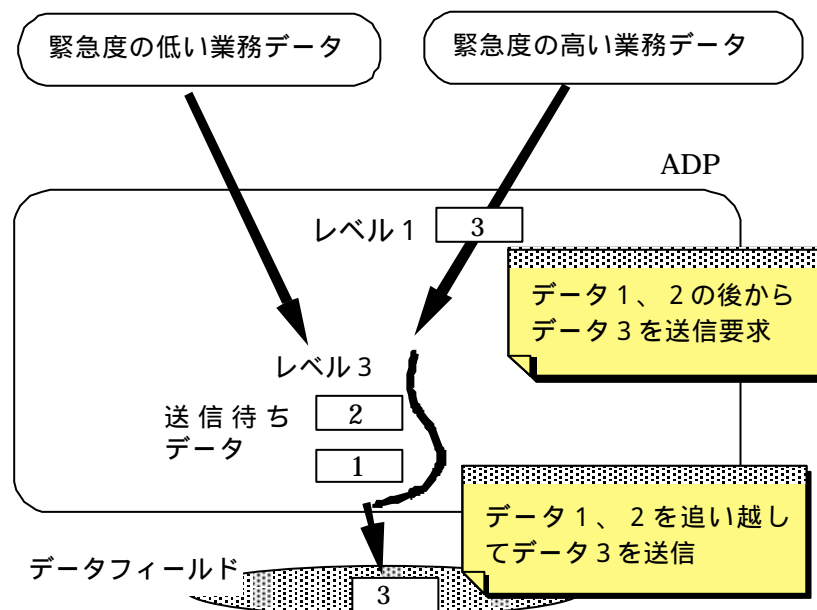


図29 メッセージの優先制御 送信側処理

4.6.3.2 受信側処理

- ・受信ノードでは、受信したメッセージのADPヘッダ内のメッセージ優先レベル（PRI）を確認し、メッセージ優先レベルが1～7である場合は、優先レベルに応じた優先でメッセージを受信しAPに受け渡す。
- ・優先レベル0のメッセージを受信した場合は、あらかじめ優先レベル0のメッセージを受信処理する優先度を受信ノード毎に設定しておき、設定してある優先レベルに受信ノードで変換してそのメッセージを処理する。通常、そのデータフィールドで使用できる最低優先のレベルで処理する。

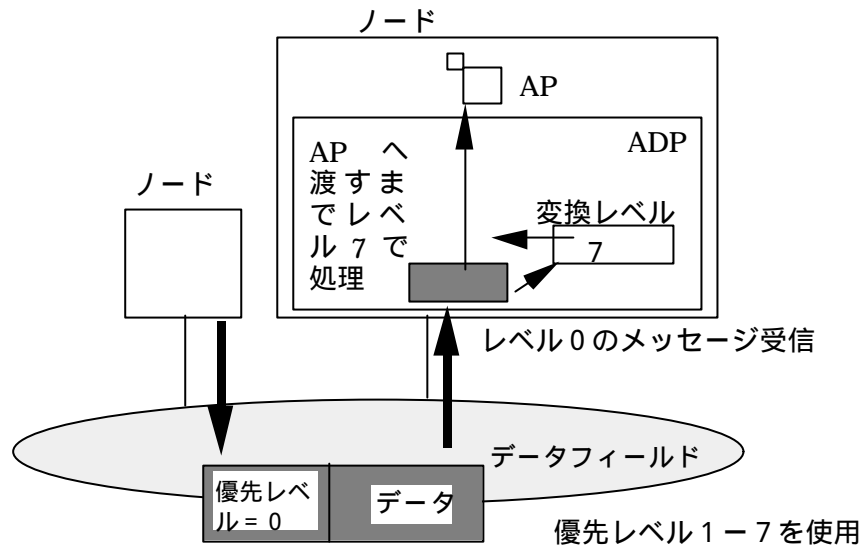


図30 メッセージ優先制御 受信側処理
優先レベル0時の優先レベル変換

4.7 メッセージ通番管理

4.7.1 機能

受信側でメッセージの抜けを検知するために、メッセージ送信通番と通番バージョン番号を使用する。

4.7.2 メッセージ送信通番

メッセージ抜けを検知するためにメッセージに付与する順次番号。

4.7.3 メッセージ送信通番バージョン番号

メッセージ送信通番のリセットを検知するために使用する情報。その内容は、メッセージ送信通番の初期化毎に獲得できる一意の値である。

[参考]

送信側ノードがダウンして通番がリセットされると、受信側ノードでは誤って通番抜けと認識する。この問題を解決するためにメッセージ通番バージョン番号を用いる。メッセージ通番バージョン番号には、タイムスタンプの使用を推奨する。メッセージ通番バージョン番号にタイムスタンプを使用した場合で、その役割を図31に示す。

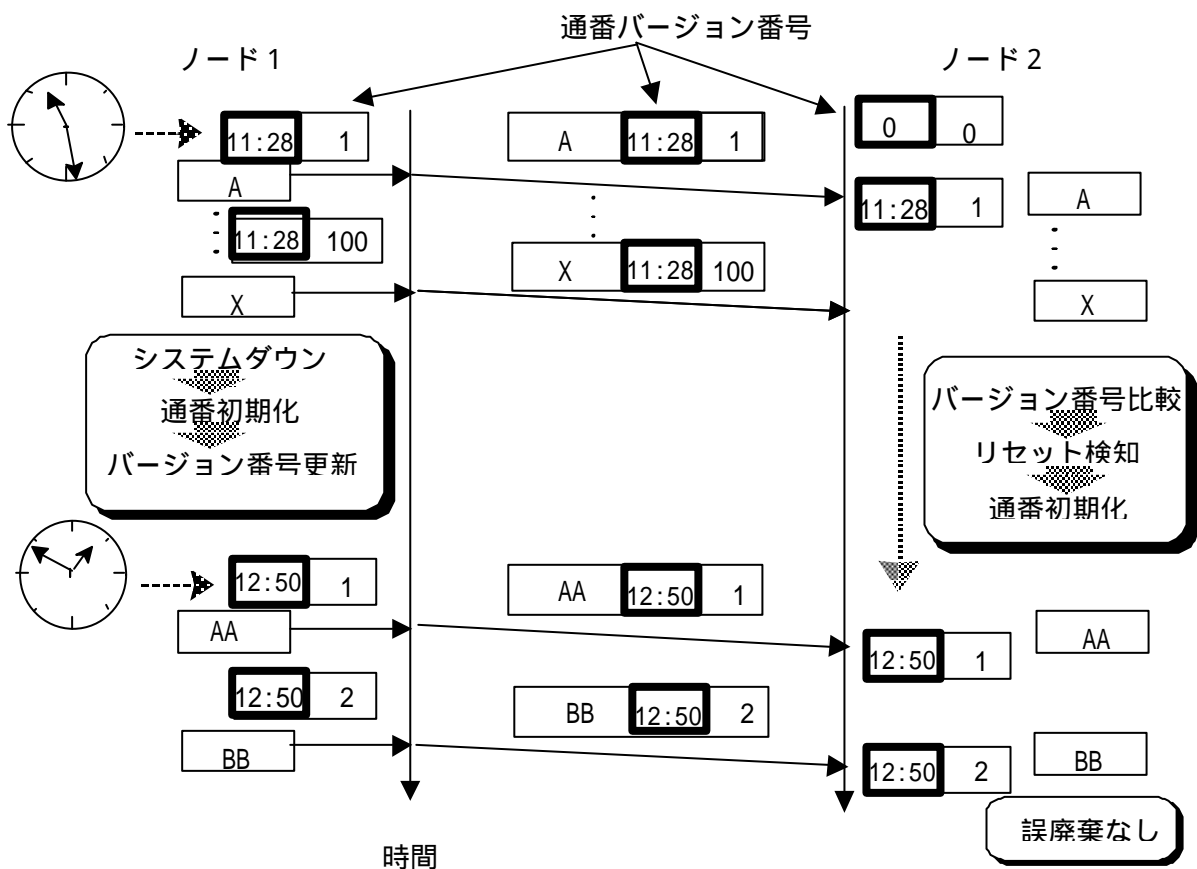


図31 通番バージョン番号

4.7.4 メッセージ通番管理処理

4.7.4.1 送信側処理

送信ノードでは、メッセージ送信通番とメッセージ送信通番バージョン番号を記憶して管理し、メッセージ送信時にそれぞれを、そのメッセージのPDUのADPヘッダのメッセージ送信通番 (SEQ) とメッセージ送信通番バージョン (V_SEQ) に格納し送信する。

4.7.1.2 受信側処理

受信ノードでは、最新受信通番とともに、最新受信通番バージョン番号をメッセージ送信元ノードごとに記憶しておく。受信メッセージのPDUのADPヘッダのメッセージ送信通番 (SEQ) とメッセージ送信通番バージョン (V_SEQ) を確認し、受信メッセージの通番バージョン番号が最新の通番バージョン番号と異なる場合は、通番がリセットされたと判定し、最新受信通番を強制的に受信メッセージの通番に一致させる。

受信メッセージの通番バージョン番号が最新の通番バージョン番号と一致していて、受信通番が飛んでいた場合は、通番抜けと判定する。

マルチキャスト通信、および1対1通信についての通番と通番バージョン管理処理の詳細は『付属書D.3』を参照。

4.8 メッセージの分割・組み立て

4.8.1 機能

TCP/UDP、IPプロトコルヘッダ、およびADPヘッダおよびメッセージを全て含んだサイズがMTUよりも大きい場合、メッセージ送信側では、ひとつのメッセージを複数のPDUに分割し送信を行う。メッセージ受信側では、分割されたPDUを受信し、元のひとつのメッセージに再組み立てを行う。

4.8.2 メッセージ分割・組み立てに使用する情報

メッセージの分割、組み立てに使用する情報を示す。

CBN : カレントフラグメントPDU番号 (先頭PDU番号は1から)

TBN : トータルフラグメントPDU数

BSIZE : PDUサイズ

ML : メッセージ長 + 64 (バイト)

SEQ : メッセージのシーケンス番号

送信ノードは、これらの情報をそのメッセージのPDUのADPヘッダ部に格納送信する。受信ノードは、そのメッセージのPDUのADPヘッダ部を確認し、メッセージの組み立てを行う。

1対1通信におけるメッセージ分割・組み立て処理の詳細は『 付属書D.4 』を参照。

4.9 連想配列プロトコル

4.9.1 メッセージ構造

メッセージを構成するデータ項目に関する情報と、データ項目の値を一体化してアプリケーションデータを構成する。

データ項目に関する情報は名称や型などがありますが、本プロトコル案ではエンコーディング規則としてASN.1(IEC8824/8825)、基本データ型としてIEC61131-3規定のデータ型を採用する。

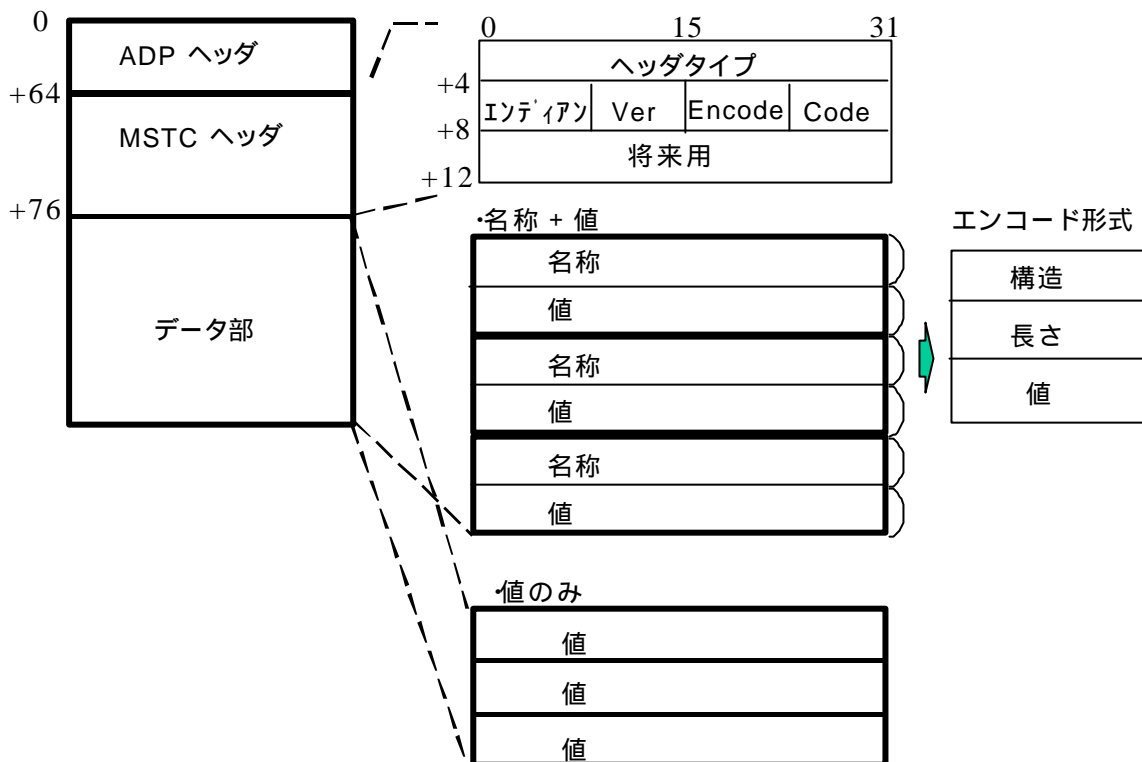


図32 メッセージ構造

MSTCヘッダ : データ部の構造を示す。ビッグエンディアンで設定する。

表4 ヘッダフォーマット

項目名	説明
ヘッダタイプ	ヘッダ識別子。ASCII表記。
エンディアン	“MSTC”固定 データ部中の、データ項目値部分のエンディアン。 1：リトル、2：ビッグ、3：なし
Ver	本プロトコルのバージョン。1 固定
Encode	データ部の利用方法を示す。 1：データ値のみ - 従来型 2：データ値及び項目情報 - 項目説明一体型（本案）
Code	データ部に用いられている文字列のコード体系を示す。

4.9.2 構造の表現

4.9.2.1. 基本データ型：詳細はIEC61131-3参照

表5 基本データ型一覧

データタイプ名	データタイプコード(16進)	説明
BOOL	C1	論理値 (TRUE/FALSE)
SINT	C2	符号付き8ビット整数値
INT	C3	符号付き16ビット整数値
DINT	C4	符号付き32ビット整数値
LINT	C5	符号付き64ビット整数値
USINT	C6	符号無し8ビット整数値
UINT	C7	符号無し16ビット整数値
UDINT	C8	符号無し32ビット整数値
ULINT	C9	符号無し64ビット整数値
REAL	CA	32ビット浮動小数点値
LREAL	CB	64ビット浮動小数点値
STIME	CC	同期時間の情報
DATE	CD	日付
TIME_OF_DAY	CE	時刻
DATE_AND_TIME	CF	日付及び時刻
STRING	D0	8ビット文字列
BYTE	D1	8ビット列
WORD	D2	16ビット列
DWORD	D3	32ビット列
LWORD	D4	64ビット列
STRING2*	D5	2バイト文字列
FTIME*	D6	持続時間 (高分解能)
LTIME*	D7	持続時間 (長精度)
ITIME	D8	持続時間 (短精度)。INT型と同じ
STRINGN*	D9	Nバイト文字列
SHORT_STRING*	DA	8ビット (1バイト長さ標識)

*は、IEC61131-3範囲外の拡張仕様。

4.9.2.2. 構造体・配列の表現

構造体、配列については、各構成要素項目毎に名称と値を持たせた構造と、要素毎に名称を付けなくともよい型を設ける。

表6 構造体・配列の表現

データタイプ名	データタイプコード(16進)	説明
ANY_STRUCT	E0	構成要素各々に、名称(含型・長さ)と値(含型・長さ)の組を含む構造体または配列
SEQ_STRUCT	E1	構成要素各々について、値(含型・長さ)のみを含む構造体または配列

4.9.3 エンコード例

4.9.3.1. 基本データ型データ項目：名称と値を持つ。

例) UINT item = 5;

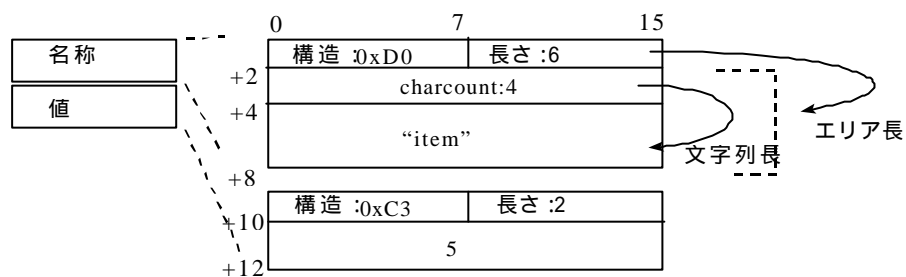


図33 エンコード例 1

4.9.3.2. 時間属性を持つデータ：1つの名称と、値及び時刻情報を持つ。

例) ANY_MAGNITUDE ::= {
 ANY_NUM
 TIME_OF_DAY
 }

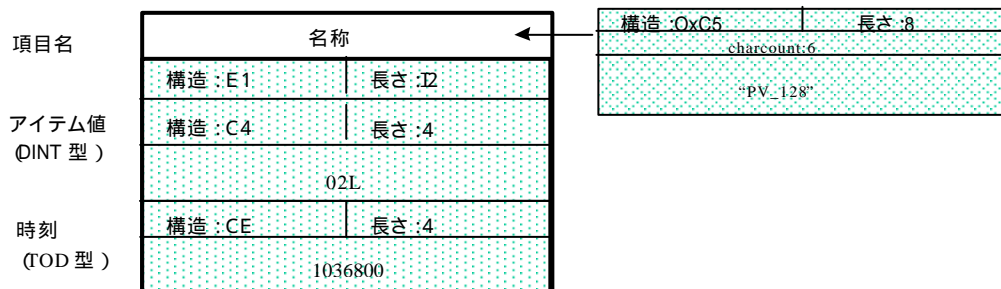


図34 エンコード例 2

4.9.3.3. 文字配列

例)

STRING str2[2]

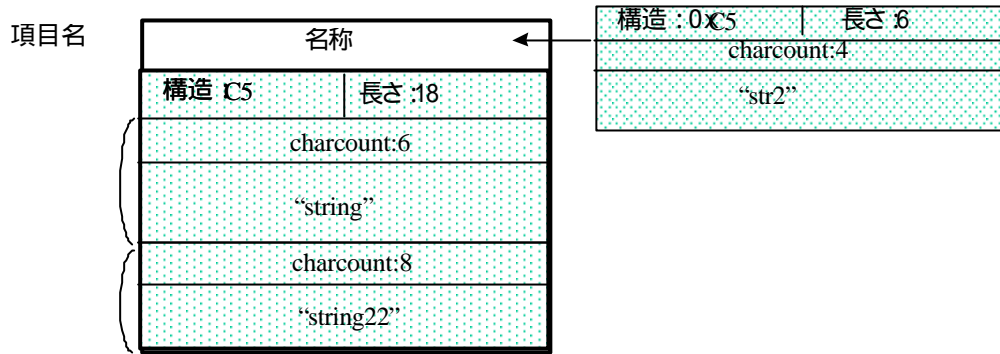


図35 エンコード例 3

4.9.3.4. 配列・構造体データ項目

例 : struct { DINT item01; DINT item02[2]; } struct1 = { 01L, { 02L, 03L } };

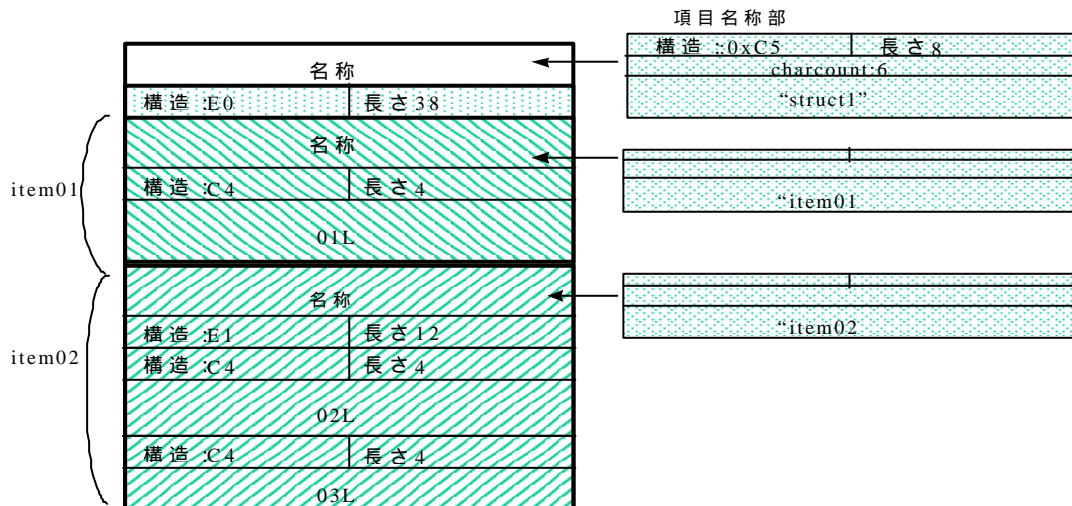


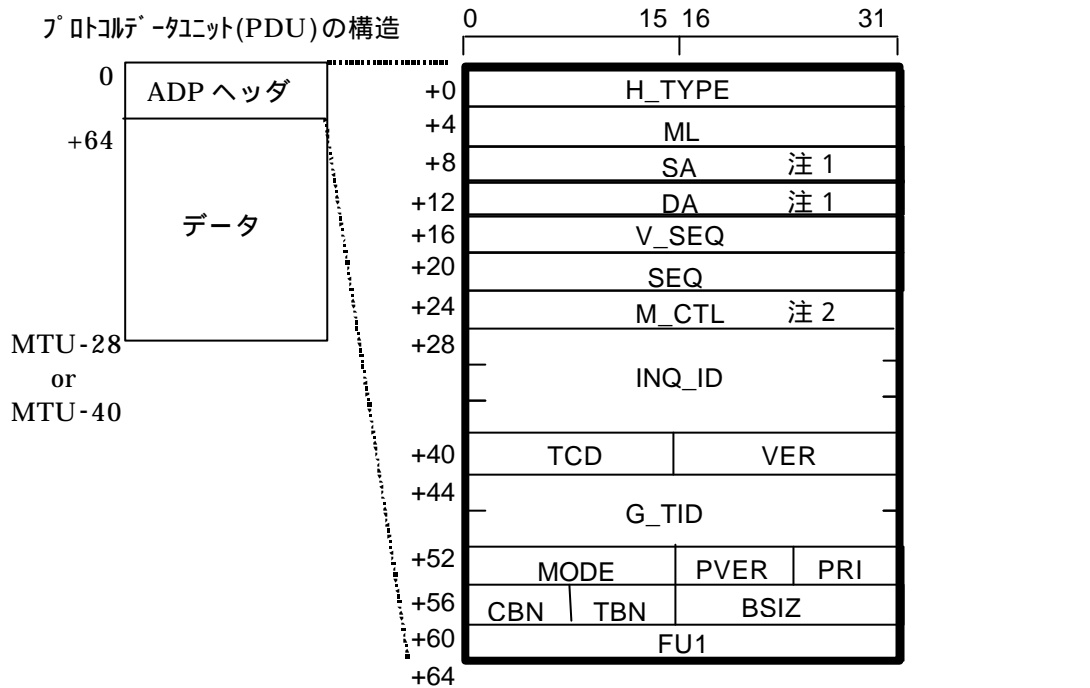
図36 エンコード例 4

5章 PDUの構成及びPDUの符号化

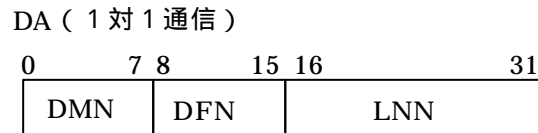
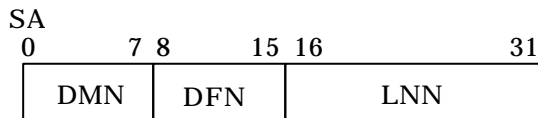
5.1 PDU構造

図37にプロトコルデータユニット (PDU) の構造を示す。PDUは固定長のADPヘッダ部と可変長のデータ部からなる。表7 にADPヘッダ部の情報を示す。

また、メッセージ、ヘッダの相対番地は、ネットワーク上を流れるデータ位置である。ビット位置も同様にネットワーク上へ流れる順に0から31ビットで表現している。以降のメッセージフォーマット、各種ヘッダ表現において同様に表記する。自律分散プロトコルでは、バイトオーダはビッグエンディアンである。

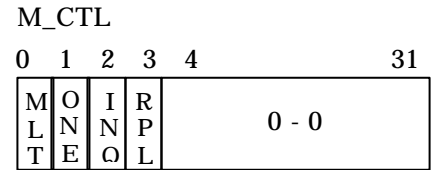


注 1 :



- (1) DMN : ドメイン番号 (現在は0固定)
- (2) DFN : データフィールド番号
- (3) LNN : 論理ノード番号(マルチキャスト)通信時のSA、1対1通信時のDA, SAで使用)
- (4) MGN : マルチキャストグループ番号(マルチキャスト通信時のDAで使用)

注 2 :



- (1) MLT : マルチキャストフラグ
- (2) ONE : 1対1フラグ
- (3) INQ : 問い合わせフラグ
- (4) RPL : 応答フラグ

図37 PDU構造とADPヘッダ

表7 ADPヘッダ部の情報

No	記号名称	サイズ	内容
1	H_TYPE	4	ヘッダタイプ。ASCIIにて"NUXM"をセット。他プロトコルメッセージとの識別のためにセット。(固定)
2	ML	4	メッセージ長 + 64 (バイト)
3	SA	4	メッセージ送信元アドレス。図37 注1参照。
4	DA	4	メッセージ送信先アドレス。図37 注1参照。
5	V_SEQ	4	メッセージ送信通番バージョン番号。通常、メッセージ送信通番が初期化された時刻をセットする。以後、“通番”と略す場合は、“メッセージ送信通番”を指す。
6	SEQ	4	メッセージ送信通番。有効値は0x00000001 ~ 0x7FFFFFFFでサイクリック使用。
7	M_CTL	4	メッセージ送信制御情報。図37 注2参照。
8	INQ_ID	12	問合せ応答識別子(将来用のため0固定)
9	TCD	2	トランザクションコード。
10	VER	2	プログラム入れ替え用プログラムバージョン番号。(将来用のため0固定)
11	GTID	8	トランザクション識別子。(将来用のため0固定)
12	MODE	2	メッセージモード(0:オンラインモード,1:テストモード)
13	PVER	1	プロトコルバージョン(1固定)
14	PRI	1	メッセージの優先レベル(1~7の値で1が最優先。優先制御未実装時0固定)
15	CBN	1	カレントフラグメントPDU番号(1以上)
16	TBN	1	トータルフラグメントPDU数(1以上)
17	BSIZE	2	PDUサイズ(PDUのサイズでヘッダサイズ含む)
18	FUI	4	将来用のため0固定。

5.2 クラス-Base-1 (マルチキャスト通信) PDU

図38 にマルチキャスト通信PDUのフォーマット、およびADPヘッダ部の設定値を示す。

データ部には通信データを格納する。

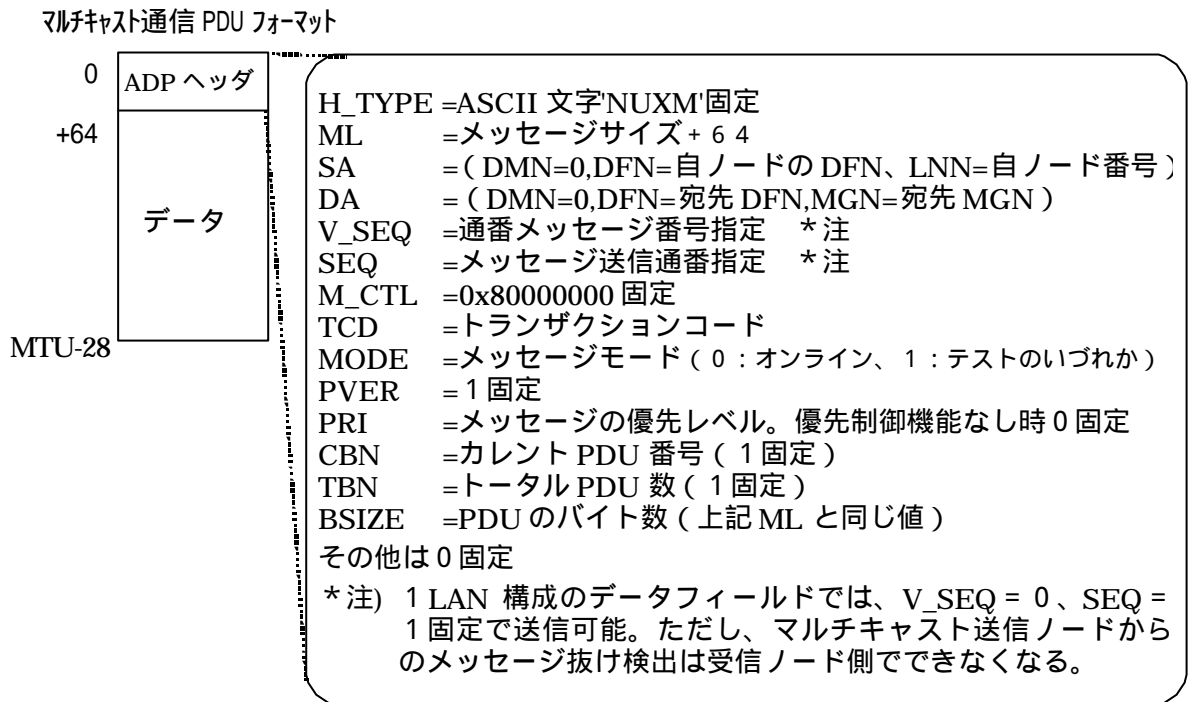


図38 マルチキャスト通信PDUフォーマットとADPヘッダ部の設定値

5.3 クラス-Base-2 (生存信号) PDU

生存信号PDUのフォーマットを図39 に示す。PDUのデータ部は、固定長の生存信号ヘッダ部と可変長の障害情報部から成り、生存信号情報は生存信号ヘッダ部に格納する。

生存信号PDUのADPヘッダ部の設定値、及び生存信号ヘッダ部の構造もあわせて図39 に示す。

生存信号ヘッダ情報を 表8 に示す。

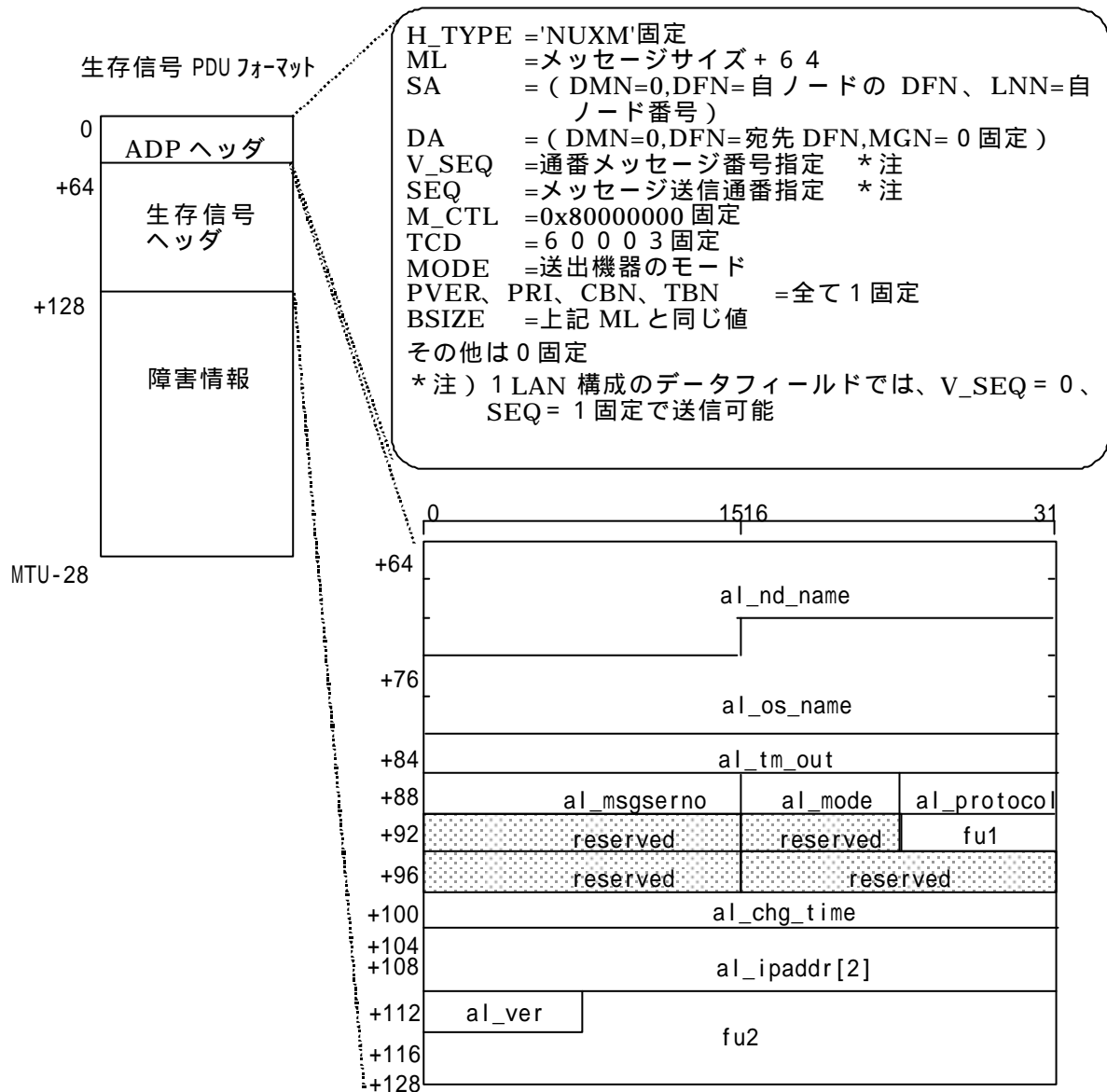


図39 生存信号メッセージフォーマットと生存信号ヘッダ

表8 生存信号ヘッダ説明

No.	記号名称	サイズ	内 容
1	al_nd_name	10	ノード名称（最後にNULLがセットされたASCII文字列で、最大9文字）
2	al_os_name	10	ベンダ機器名称（最後にNULLがセットされたASCII文字列で、最大9文字）以下のガイドラインに従うことが、望ましい。 “BN_MN” BN : ベンダー名コード MN : 各ベンダーで定めた機器名称 もしくはOS名称（最大6文字） 付属書B.にベンダ名コード一覧を記載。
3	al_tm_out	4	生存信号タイムアウト監視時間。生存信号を送信したノードの生存信号が未到達となってからそのノードを死状態と認識するまでの時間。（秒単位）
4	al_msgserno	2	生存報告メッセージ通番。本パラメータは二重化 LAN 制御においてのみ使用します。通番の有効範囲は、1~0x7fffでサイクリックに使用します。二重化 LAN 構成をサポートしていない機器では、0を指定。
5	al_mode	1	生存報告モード。本仕様の自律分散プロトコルでは、“1”での生存信号送信が必須。 1 : 通常（生状態） 通常は、この値で出力する。 2 : shutdown予告 機器が停止することによる生存信号停止であることを他の機器に知らせる目的で、生存信号送出を止める直前に、このモードで送出する。このモードでの送出はオプション。 3 : メンテナンス予告 機器をメンテナンスするための生存信号停止であることを、他の機器に知らせる目的で、生存信号送出を止める直前に、このモードで送出する。このモードでの送出はオプション。
6	al_protocol	1	生存信号ヘッダ以降のデータ部の用途種別を指定。本仕様の自律分散プロトコルでは“4”を設定する。 1~3 : リザーブ（既存の他プロトコルで使用） 4 : 本仕様の自律分散プロトコル、 5~ : リザーブ（将来用）
7	fu1	1	将来用で0固定。
8	al_chg_time	4	ノード状態変化発生時刻。グリニッジ時間（1970年0時0分0秒からの経過秒）にて年月日時分秒をセットすることを推奨。グリニッジ時間を設定できない場合は、0を指定。 ノード状態変化は以下の場合に発生する。この値はノード状態表示で使用。 （1）al_mode=1 で生存信号送出を開始（停止から生存状態になった時点） （2）al_mode=2 で生存信号送出（Shutdown（機器停止）要求受け付け時点） （3）al_mode=3 で生存信号送出（メンテナンスによる機器停止受け付け時点）
9	al_ipaddr[0] al_ipaddr[1]	4	al_ipaddr[0]は、LANのIPアドレスを設定する。al_ipaddr[1]には、0を設定する。
10	al_ver	1	生存報告メッセージバージョン。（現在は1固定）
11	fu2	15	将来用で0固定。
12	reserved	7	0を設定。

5.4 クラス-Opt-2-a (障害情報)

障害情報は、生存信号PDU (5.3を参照) の障害情報部に格納する。

障害情報部の構造を図40に障害情報の内容を表9 に示す。

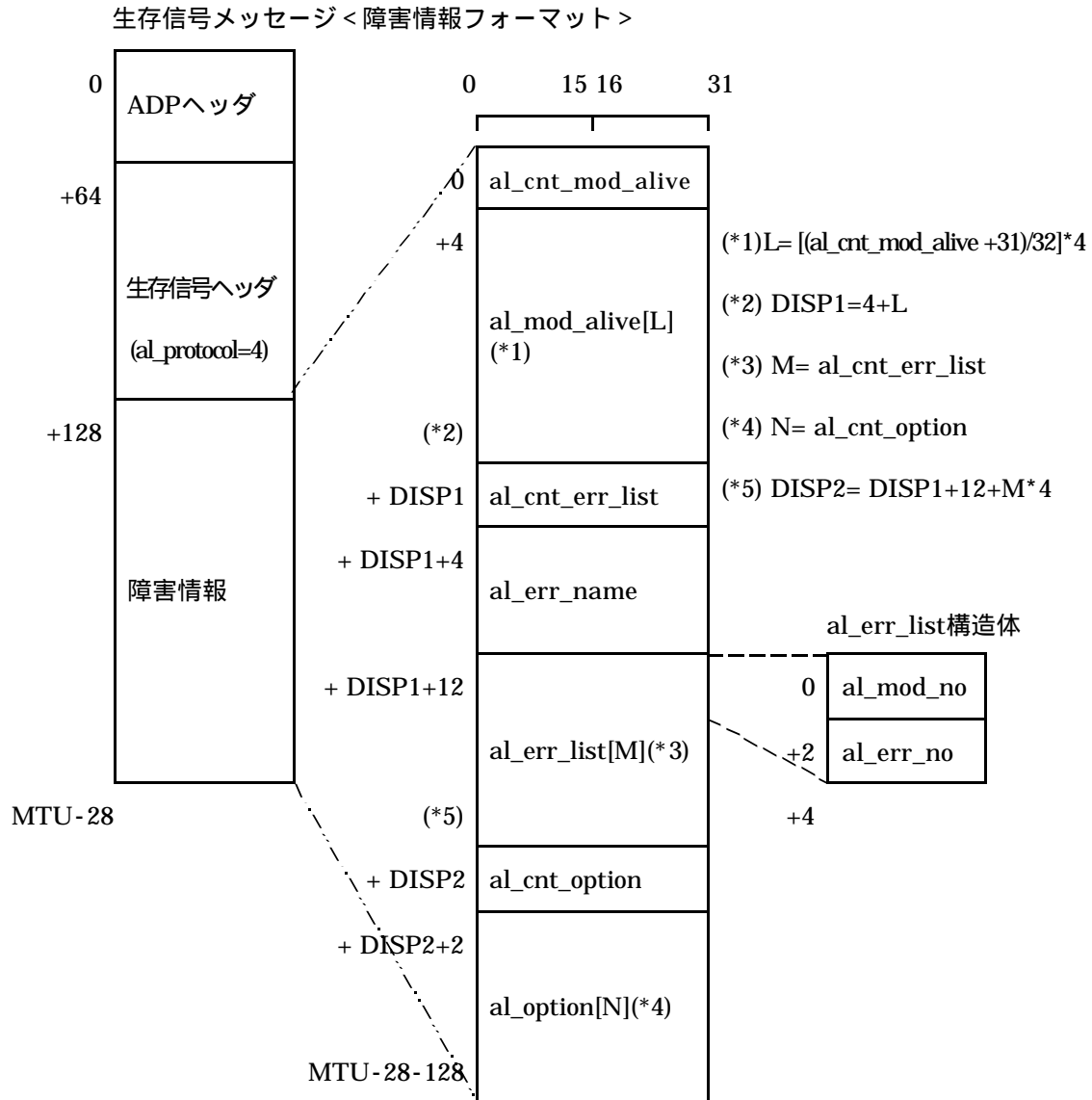


図40 障害情報フォーマット

表9 障害情報の内容

No	記号名称	サイズ	内 容
1	al_cnt_mod_alive	4	生存報告モジュール(*4)数
2	al_mod_alive[L]	L (*1)	<p>モジュール(*4)生存情報</p> <p>(1) モジュール番号(*5) X (1~) が「生」のとき al_mod_alive[(X-1)/8] の第 X-1- [(X-1)/8]*8 ビットを クリア (0)</p> <p>(2) モジュール番号(*5) X (1~) が「死」のとき al_mod_alive[(X-1)/8] の第 X-1- [(X-1)/8]*8 ビットを セット (1)</p>
3	al_cnt_err_list	4	報告エラー数。最大は、[(MTU-28-128-4-4-8) / 4]。(*2)
4	al_err_name	8	エラー名称。エラー番号体系毎に付けた名前。 この名前は、監視ツールがエラー番号に対応するエラーメッセージを 検索するエラーメッセージファイル名称のプリフィックスとして使用 する。
5	al_err_list[M]	M*4 (*2)	エラー情報のリスト
6	al_mod_no	2	モジュール番号(*5) (1~)。
7	al_err_no	2	エラー番号。al_err_nameで体系化された、al_err_nameで一意的な番号。
8	al_cnt_option	4	オプション情報バイト数
9	al_option[N]	N (*3)	オプション情報 (フォーマットを規定しない自由に使ってよい領域)

(*1) $L = [(al_cnt_mod_alive + 31) / 32] * 4$

(*2) $M = al_cnt_err_list$

(*3) $N = al_cnt_option$

(*4) モジュール：ボードなどのハードウェアやAPなどの状態や障害発生を認識したい
部位の単位。

(*5) モジュール番号：モジュールを識別する番号。

5.5 クラス-Opt-3 (1 対 1 通信) PDU

図41に 1 対 1 通信PDUのフォーマット、およびADPヘッダ部の設定値を示す。

データ部に通信データを格納する。

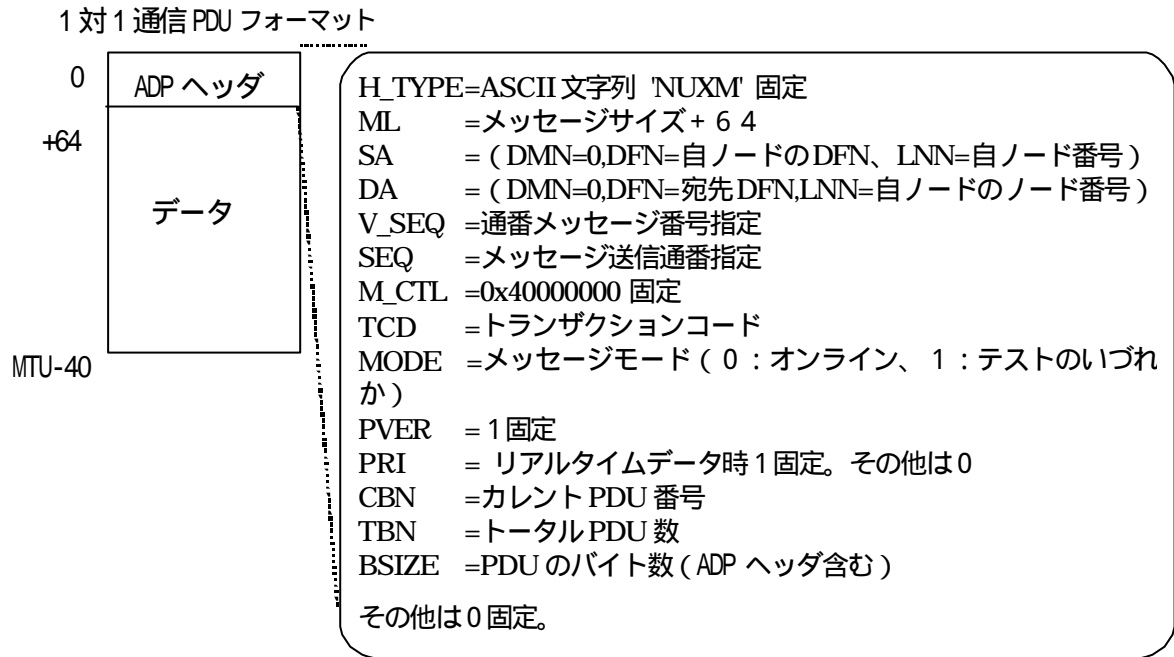


図41 1 対 1 通信PDUフォーマットとADPヘッダ部の設定値

6章 適合性

自律分散プロトコルの仕様に適合する実装のための適合性要求を示す。

6.1 適合性要求事項

自律分散プロトコルを提供するものは表10 に示す適合性要求で通信機能を実装しなければならない。かつ、各通信機能は表10 に示す適合性要求で共通項目を満たした実装を行わなければならない。

「必須」の印のある機能/項目については必ず実装しなければならない。「選択」の印のある機能/項目については全く実装しないことも、複数機能/項目実装することも可能であるが、実装している機能/項目を製品に明示しなければならない。

表10 通信機能における適合性要求

機能	クラス	規定項	適合性要求	
			送信	受信
マルチキャスト通信	Base - 1	4.1	必須	必須
生存信号	Base - 2	4.2	必須	選択
障害情報	Opt - 2 a	4.3	選択	選択
1対1通信	Opt - 3	4.4	選択	選択

表11 各通信機能ごとの基本項目における適合性要求

項目	規定項	適合性要求					
		マルチキャスト通信		1対1通信		生存信号	
		送信	受信	送信	受信	送信	受信
テスト支援	4.5	必須	必須	必須	必須	必須	規定しない
メッセージ優先制御	4.6	必須	選択	必須	選択	必須 (1固定)	選択
メッセージ通番管理	4.7	選択	選択	必須	必須	選択	選択
メッセージの分割組み立て	4.8	規定しない		選択		規定しない	
最大メッセージ長	-	必須 ((MTU-92)バイト)		必須 (16Kバイト)		必須 ((MTU-92)バイト)	
連想配列	4.9	選択		選択		規定しない	

付属書 A . システム用TCD

[参考]

表12 システムトランザクション一覧

TCD	名称	意味 / 内容
60000 ~ 60002	リザーブ	本書の規定範囲外の特別な監視 / 制御プログラムによりリザーブされています。
60003	生存信号 トランザクション	データフィールドに属するノードが生存信号として送信し、他ノードにて監視するための TCDです
60004 ~ 60007	リザーブ	本書の規定範囲外の特別な監視 / 制御プログラムによりリザーブされています。
60008 ~ 60015	リザーブ	将来用にリザーブされています。
60016 ~ 60059	リザーブ	本書の規定範囲外の特別な監視 / 制御プログラムによりリザーブされています。
60060 ~ 65399	リザーブ	将来用にリザーブされています。
65400 ~ 65534	リザーブ	本書の規定範囲外のコントローラ内部通信用としてリザーブされています。

付属書 B . ベンダー名コード一覧

ベンダー名コード(BN)は、生存信号ヘッダー内 al_os_name の中のベンダー名を表わす文字列である。

表13 ベンダー名コード一覧

BN	ベンダー名
AB	アレンブラドリージャパン(株)
ABB	ABB Industry
DS	(株)大同信号
HI	(株)日立製作所
KS	(株)京三製作所
ME	(株)三菱電機
NS	(株)日本信号
OM	オムロン(株)
OMR	
SE	(株)正興電機製作所
SS	セイコープレシジョン(株)
TD	東洋電機製造(株)
TK	東京機械(株)
TO	(株)東芝
YE	(株)横河電機

ベンダー名コードは財団法人製造科学技術センター（MSTC）が管理する。新たにベンダー名コードを確保したい場合は、MSTCに申し入れを行う。

付属書 C . 実装上の注意事項

C.1 メッセージ長

本仕様で規定するマルチキャスト通信の最大メッセージ長は、1408バイトで、1対1通信の最大メッセージ長は、16Kバイトである。

[参考]

マルチキャスト通信の最大メッセージ長は、各機器のUDPブロードキャスト最大送受信バイト数 (MTU) により制限を受けることがある。

$$\begin{aligned} \text{マルチキャスト通信の最大メッセージ長} &= \text{MTU} - \text{IP/UDPヘッダ (28)} - \text{ADPヘッダ (64)} \\ &= 1408 \text{バイト} \end{aligned}$$

イーサネットのMTUが1500であっても機器により内部バッファの管理のしかたにより最大転送バイト数がこの値よりも小さい場合がある。このような機器を使用する場合は、その機器で扱える最大ユーザメッセージ長をマニュアル、取り扱い説明書等へ明確に記載しなければならない。また、メモリ容量の制限により、バッファサイズを1408バイト以上のバッファを複数所有するのが難しい機器においては、256、512、1024バイトのバッファサイズの中から選択できるようにすることを推奨する。この場合もその機器で扱える最大ユーザメッセージ長をマニュアル、取り扱い説明書等へ記載する必要がある。

C.2 TCPコネクション管理

C.2.1 コネクションの補足制御情報

通信を行うノードのTCP機能にその機能が存在する場合は、設定しなければならない。

(1) KEEPALIVE機能

コネクションは、相手ノードがクラッシュし突然停止する場合を考慮し、相手ノードの状態を監視する目的でKEEPALIVE機能を持つノードでは、KEEPALIVE機能を有効にする必要がある。

送信するノードは、上記異常を送信タイムアウトで検知できるが、受信待ちだけしている（送信がある一定期間ないとき）ノードでは検知できない。この機能を使用することで相手ノードが異常停止したことをタイムアウトで検知できる。

(2) TCPウィンドウサイズの調整

TCPのコネクションを確立する双方のノードにおいて、TCPウィンドウサイズ（コネクション単位の送受信バッファサイズ）を同じにしなければならない。TCPのコネクション間でのウィンドウサイズが同じでない場合通信スピードが著しく低下する。ノード間のTCPウィンドウサイズを同じに設定できるようになっていなければならない。ウィンドウサイズを変更できないノードとコネクションを接続する場合は、もう一方のウィンドウサイズを相手と同じサイズにする必要がある。

各機器は、自分のデフォルトウィンドウサイズが何バイトか、ウィンドウサイズを変更できる方法をマニュアル等に明確にする必要がある。

C.2.2 1対1通信経路異常の検知

1対1通信で送受信している最中に、相手ノードの停止、異常発生、あるいはLANの障害が発生し、TCPコネクションが切断されたことを検出する機能を有しなければならない。このようなTCPコネクションが切断された場合は、1対1通信経路異常としてそのTCPポートは解放（close）しホールド状

態にする。この検知は、LAN通信アダプタからそれら障害を検知するケースと、TCPの送信タイムアウト、あるいはKEEPALIVEによるタイムアウトにて検知する。また、障害が発生したことを保守員に通知できるようエラー表示、エラーログ等の異常処理を組み込む必要がある。

C.2.3 その他注意事項

- TCPでは、コネクションを確立時にお互いの扱える最大TCPセグメントサイズを通知し、小さいほうのサイズで以後のデータ転送を行う。この最大TCPセグメントサイズは、機器により異なる。機器1が1350バイトで、機器2は1024バイトとするとTCPセグメントサイズは1024バイトとなり、TCP送受信の単位になる。TCPセグメントには、ADPヘッダを含んだメッセージ、あるいはパケットが格納される。TCPセグメントが小さいと転送効率が悪くなる。各機器は、最大TCPセグメントをドキュメントに明記する必要がある。
- 各機器は、TCPのウィンドウサイズを変更できるか否か、および変更できるサイズの範囲をドキュメントに明記する必要がある。
- 各機器は、立ち上げ時に自分のMACアドレスとIPアドレスを格納して、ARP request パケットを発行し他の機器のARPテーブルを初期化する必要がある。(機器の MAC アドレス変更に対応するため)

C.3 メッセージの優先制御

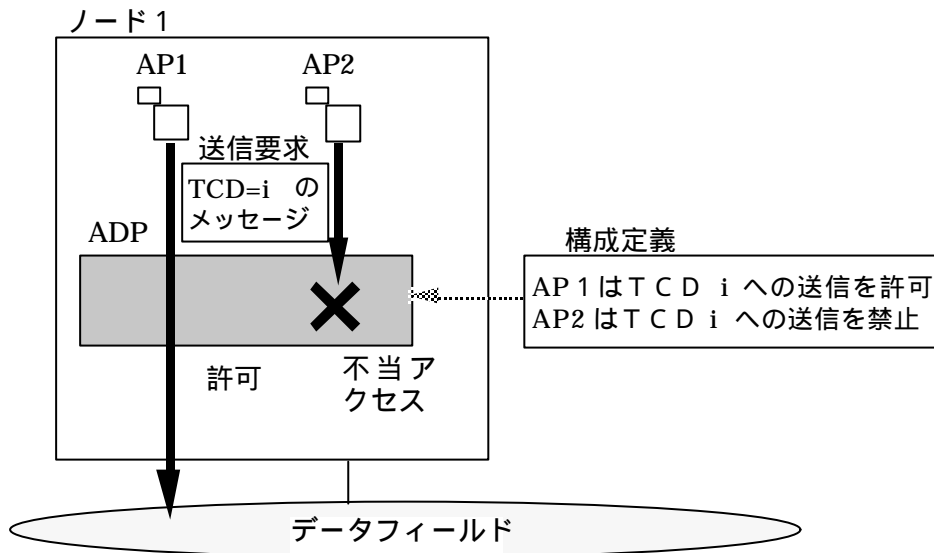
本仕様では、メッセージの優先制御機能の実装が必須であるが、優先制御を実施するデータフィールドに接続された場合に他機器との間でプロトコル異常の発生を防止するため、次の規則に従う必要がある。

- 1) 自機器の送信優先レベルが最低レベルでは問題となる場合は、予め構築時に自機器が送信するメッセージの優先レベルをいくつにするかをユーザ定義できるようにする必要がある。
- 2) ユーザシステムにて優先レベル1で送信してかまわないという保証がある場合には、優先レベル1固定で送信してもかまわない。

上記、機器の使い勝手からは上記1)をベースに実装し、ユーザ定義がない場合、優先レベルを0で送信することを推奨する。ユーザ定義があれば、定義された値を優先レベルとして送信する。

C.4 TCDアクセス権制御

APが不当なメッセージを送信したり、不当なメッセージを受信しないよう防御する機能で、APからの送信または受信メッセージのアクセス権制御を行う。アクセス権制御はTCDをチェックすることにより行う。構築時に、各AP毎に送信、受信を許可するTCD番号を定義する。送受信許可のないTCD番号に対し送信要求したり、受信要求するとアクセス違反としてその要求を拒絶する。



C.5 ノード内状態変化の統計情報とログ

自律分散プロトコルを実装した製品には、送受信中に通信アダプタの障害、あるいは送受信バッファのオーバーフローなどの状態変化を統計情報、イベントトレース、あるいはログし、通信異常が発生した時に迅速にトラブルシューティングできる機能を装備する必要がある。上記手段がない機器でも、『4.3 クラス-Opt-2-a (障害情報送出) 機能』に記載した障害情報の送信機能をサポートする必要がある。

ノード内状態変化イベントとしては次項目などがある。

- 通信アダプタなどのハードウェア障害
- 通信バッファの使用率、オーバーフロー
- TCP/IP、UDP/IPプロトコルの異常検知

など

C.6 拡張性

自律分散プロトコルをプラットフォームに構築したシステムでは、システムの段階的な拡張が容易に行えるように、既存のオンライン処理を停止することなく、ノード、マルチキャストグループなどをオンライン中に追加、削除できることが必要とされる。

自律分散プロトコルを実装する機器は、オンラインのシステムを停止することなく、新規増設、削除できるように設計する必要がある。

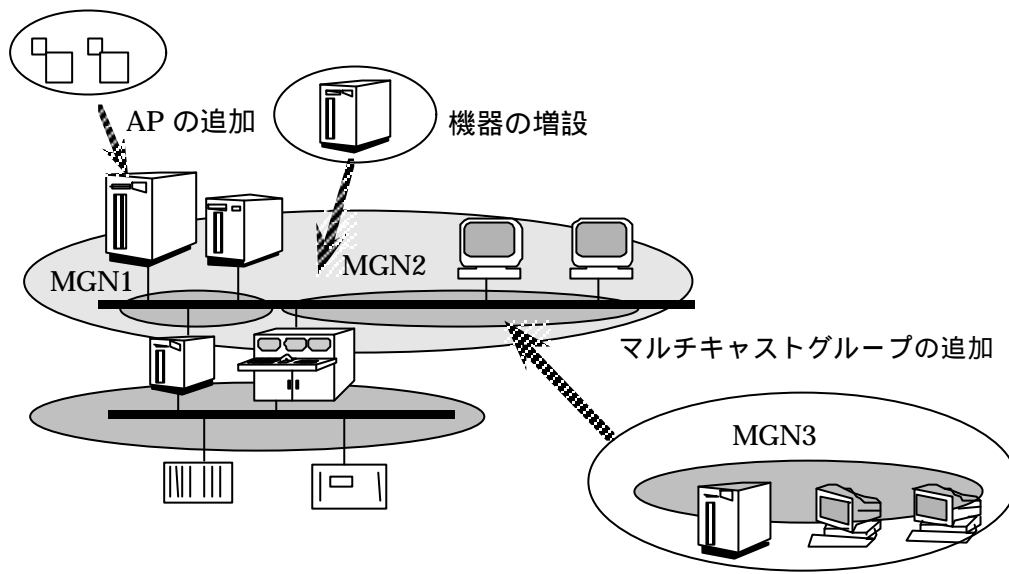


図43 システム拡張性

付属書 D . (参考) 処理手順

本仕様書は、実装例としてWindowsやUNIXで使用されるソケットの概念を用いて説明し、次のAPI名を使用している。

socket、bind、connect、sendto、recvfrom、read、write、setsockopt、select

D.1 マルチキャスト通信の送受処理

D.1.1 送信処理

次にマルチキャストメッセージの送信手順をソケットを使用する機器を例に示す。

- (a)送信先に対するADPヘッダを作成する。
- (b)メッセージにADPヘッダを付加しPDUを作成する。

[参考]

本仕様では、マルチキャスト通信のメッセージサイズが (MTU-92) より小さいためメッセージの分割は行わない。

- (c)宛先データフィールドのマルチキャスト送信ポート用ソケットへ上記(b)のPDUを送信 (sendto) する。送信sendtoの引き数は、宛先データフィールドのブロードキャストIPアドレス、宛先マルチキャストグループ番号に対応する宛先UDPポート番号を指定する。

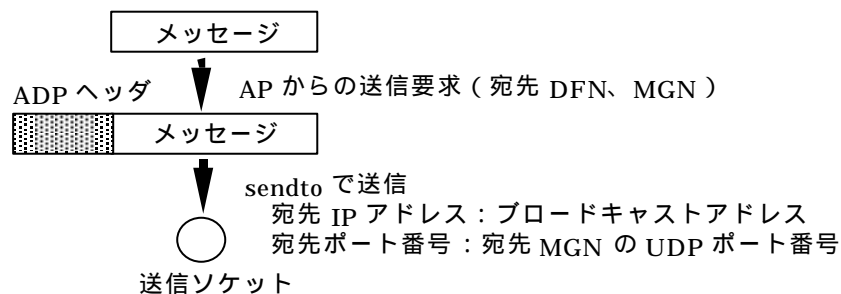


図44 マルチキャストメッセージ送信

D.1.2 受信処理

次に、ソケットを使用する機器を例に、マルチキャストメッセージの受信手順を示す。

- (a) マルチキャストグループMGNi受信用のソケットからPDUを読み込む。
- (b) 受信PDUのADPヘッダを取り出し、ヘッダの合理性チェックを実施する。ヘッダ内の有効値の範囲チェックを実施し他機器からの異常ヘッダ受信により機器異常を来たさないようにする必要がある。
- (c) ADPヘッダ内のTCDに応じて、メッセージ (PDUのデータ部) をAPに渡す。

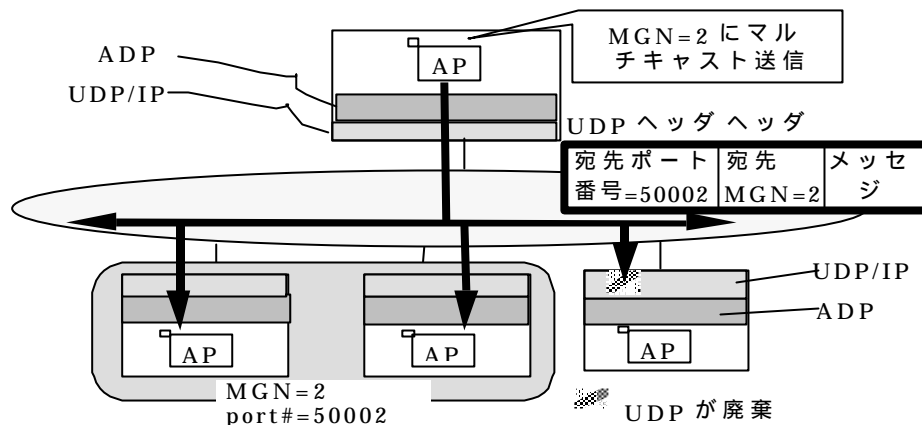


図45 マルチキャストの送受信処理概要

D.2 1対1通信の送受信処理

D.2.1 送信処理

次に1対1通信の送信手順を示す。

- メッセージサイズがLANへ送信できる(MTU-104)より小さい場合には、(c)へ。メッセージサイズが大きい場合は、メッセージを分割しパケット形式にしたのち各パケットを順次(c)で送信する。メッセージの分割については、『D.4 メッセージの分割・組み立て』にて説明する。
- メッセージ(あるいは分割したメッセージ)にADPヘッダを付加し、PDUを作成する。
- 相手ノードへのコネクションへPDUを送信します。相手ノードのウィンドウが閉じて送信ができない時は、ウィンドウが空くまで待つ。ただし、送信はできなくても受信はできなければならない。このため送信要求したパケットが全て送信完了するまで待たされる(ブロック状態)ような実装をしてはいけない。

[参考]

マルチキャスト送信ソケットを非ブロック状態にし、送信パケットが送信可能か、否か判定し、可能であればTCPへ送信することを推奨する。ソケットを使用する場合、送信できる状態でない場合はwriteはEWOULDBLOCKでリターンする。これが返る場合は、相手TCPから送信待ち要求されているため、そのTCPのソケットをselectに設定し送信できる状態になるまで他処理を実行しています。selectより送信可能を通知されると送信を再開する。

TCPは、バイトストリームのデータ転送のためひとつのPDUの送信要求を行っても、全てのデータでなく数バイトだけが送信され、他データは送信できないことが起こる。この場合は、データを重複させないように、前回送信できたデータの次のデータから送信する。

[注意]

TCPの通信は、ひとつのコネクションで全二重通信する。よって、受信あるいは送信のいずれかが待ちになりもう一方の処理ができなくなる実装は禁止する。

D.2.2 受信処理

次に1対1通信の受信手順を示す。

- (a) 相手ノードのコネクションからADPヘッダ分の64バイトを読み込む。
TCPのデータ通信は、バイトストリームであるためPDUの境界がどこかを明確に知ることができない。このためPDU境界がどこかを自分で管理する必要がある。
図46にひとつのPDUがひとつのTCPパケットで受信された場合の受信処理を、図47にひとつのPDUが複数のTCPパケットにまたがって受信された場合の受信処理を示す。
- (b) 受信したPDUのADPヘッダを取り出し、ヘッダ情報のチェックを行う。
- (c) 受信したADPヘッダのTBNとCBNの関係がTBN = CBNでない場合は、メッセージが複数のPDUに分割されているため、(a)、(b)を繰り返し分割されている数だけPDUを取り込み、メッセージの組み立て処理を行う。組み立て処理については、『D.4メッセージの分割・組み立て』にて説明する。TBN=CBNであれば完結したメッセージの受信であるため(d)へ。
- (d) ADPヘッダ内のTCDに応じて、メッセージをAPに渡す。

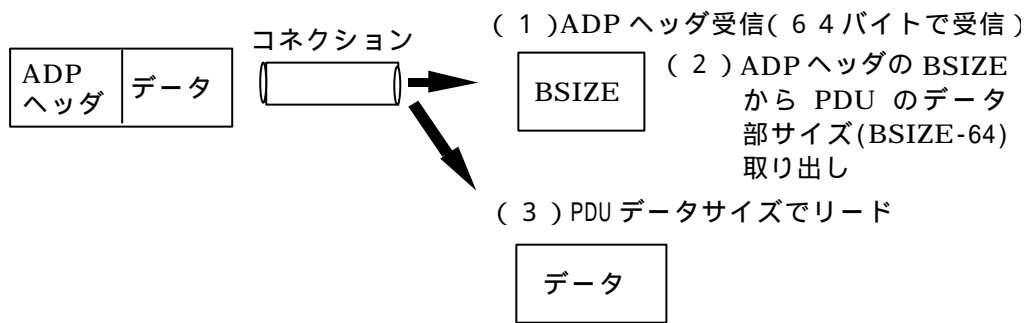


図46 ひとつのPDUがひとつのTCPパケットで受信された場合の受信処理

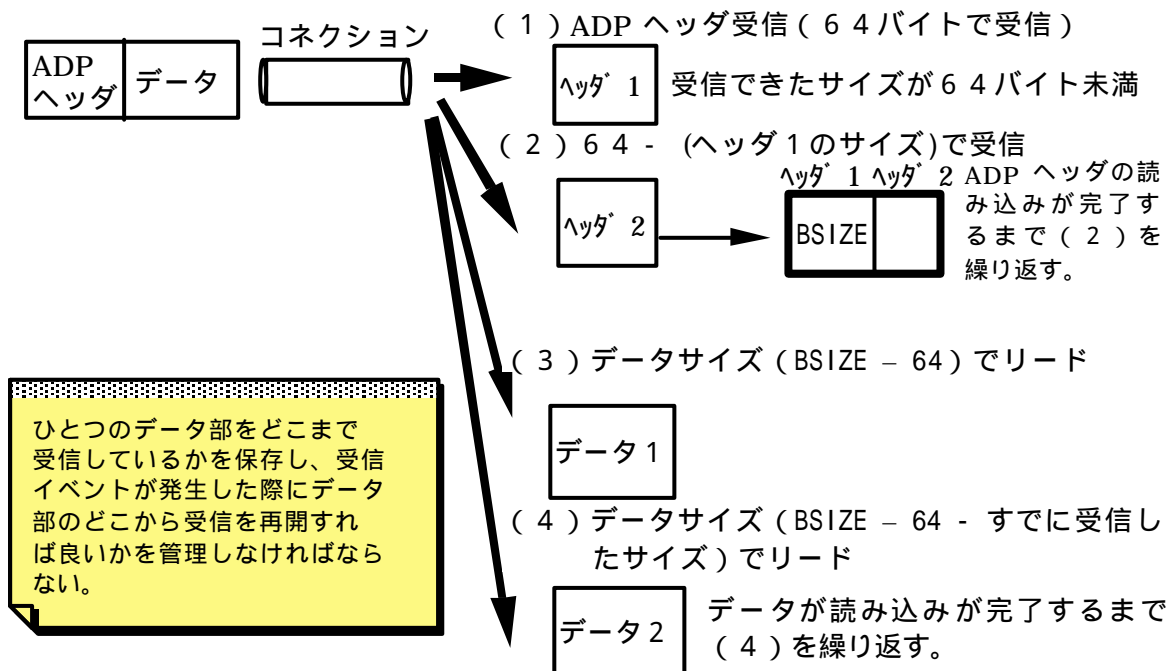


図47 ひとつのPDUが複数のTCPパケットにまたがって受信された場合の受信処理

D.3 通番と通番バージョン管理処理

D.3.1 マルチキャスト通信での処理

D.3.1.1 送信側ノードでの管理方法

< 割り当て >

- 送信通番のバージョンを管理するため、通番バージョン番号(V_SEQ)を、ノードに1つ設ける。
- 送信通番(S_SEQ)については、MGN 毎かつメッセージ優先レベル毎（レベルがひとつの時はひとつ）に管理する。
- 通番バージョン番号には、タイムスタンプをセットする。

< 初期化時の処理 >

システム立ち上げによる上記送信通番を初期化する場合は、初期化時刻を通番バージョン番号 V_SEQ にセットした後、S_SEQ 値を 1 にする。

< 送信時の処理 >

メッセージ送信時、V_SEQ を ADP ヘッダ内 V_SEQ (バージョン番号) にセットし、S_SEQ を ADP ヘッダ内 SEQ (送信通番) にセットします。この時の S_SEQ の値により次のように更新する。

- 通番 S_SEQ = 0x7FFFFFFF 時
S_SEQ の値をカウントアップ (+ 1) する。
- 通番 S_SEQ = 0x7FFFFFFF 時
カウントアップせずに S_SEQ を 1 に戻す。

[注意]

V_SEQ=0、SEQ=1 にすると受信側は通番チェックを行わない。

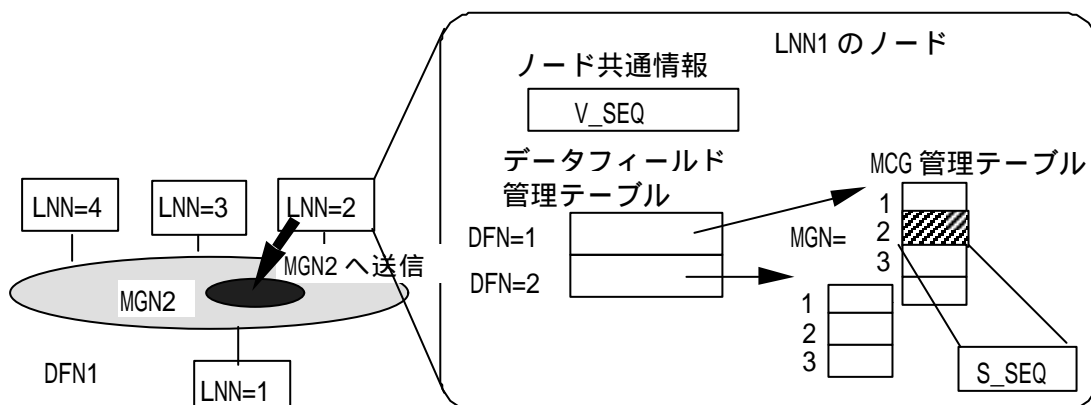


図48 マルチキャスト送信側の通番管理

D.3.1.2 受信側ノードでの管理方法

< 前提条件 >

- データフィールド毎かつメッセージ優先レベル毎に、マルチキャスト受信通番管理テーブル (MCSEQ) を持つ。
- マルチキャストメッセージ受信時、送信元ノードのDFNと優先レベルより、データフィールド管理テーブル経由で上記MCSEQにアクセスする。
- 各MCSEQは、MGN毎、LNNごとのマトリクスになっており、マトリクスの各要素は最新受信通番 (R_SEQ) と最新受信通番バージョン番号 (R_V_SEQ) がセットされる。
- メッセージ送信元LNN、及びMGN毎に受信メッセージの最新通番(R_SEQ)と通番バージョン番号 (R_V_SEQ)を保持する。

< 初期化時の処理 >

立ち上げ後、最初に受信したメッセージを無条件に受信するため、立ち上げ時の初期設定では R_V_SEQ、R_SEQ共に "0" をセットする。

< 受信時の処理 >

メッセージ受信時、ADPヘッダ内SAより送信元ノードの L N NとSEQ(送信通番)及びV_SEQ(通番バージョン番号)を読み込み、上記L N Nと受信ポート番号からMCSEQを検索し、R_V_SEQとR_SEQを求め、次ページの通番チェック処理を実行する。

[注意]

V_SEQ=0、SEQ=1のメッセージは通番チェックをせずに受信しAPへのメッセージを渡す。

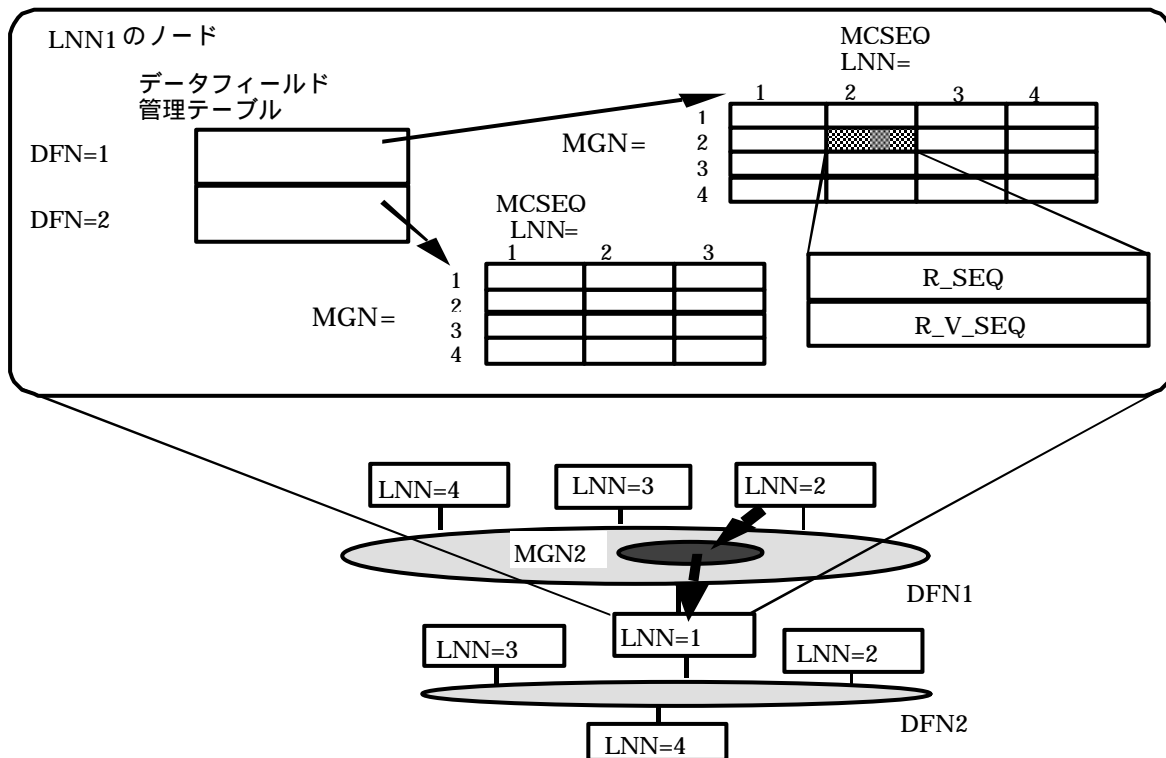


図49 マルチキャスト受信通番管理

D.3.1.3 マルチキャスト受信処理における通番チェック処理

1) R_V_SEQ = 0の場合

メッセージを無条件に受信し、V_SEQ、SEQを受信通番管理テーブル(MCSEQ)にセットする。

2) R_V_SEQ = 0、かつ、V_SEQ = R_V_SEQの場合

R_SEQとSEQを比較して、重複メッセージの検知、廃棄を行う。

3) R_V_SEQ = 0、かつ、V_SEQ < R_V_SEQの場合

送信側で通番が初期化されたと判断し、R_V_SEQとR_SEQの値をV_SEQとSEQの値に更新する。

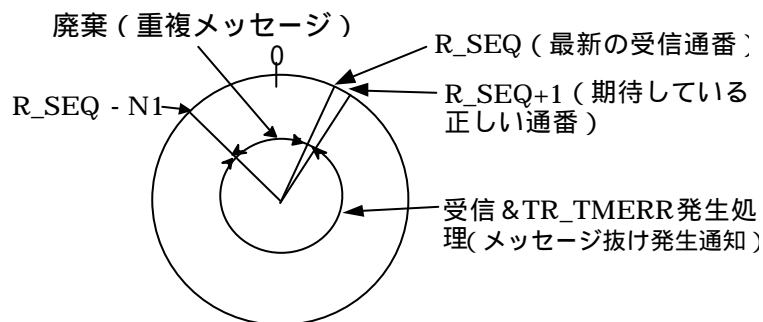
表14 通番チェックの詳細条件

No.	条 件	判定結果
1	$R_SEQ \neq 0x7fffffff \ \&\& \ R_SEQ + 1 == SEQ$	正常なメッセージ受信
2	$R_SEQ == 0x7fffffff \ \&\& \ SEQ == 1$	
3	$R_SEQ > N1 \ \&\& \ R_SEQ - N1 < SEQ \leq R_SEQ$	重複メッセージ受信
4	$R_SEQ < N1 \ \&\& \ (0 < SEQ - R_SEQ \ \ \ \ 0x7fffffff - (N1 - R_SEQ) < SEQ - 0x7fffffff)$ *	
5	上記以外	メッセージ抜け

N1: 重複メッセージと見なす通番値

* : No4の判定条件は、判定条件を簡略化するためにR_SEQとSEQに0x80000000を加え(それぞれXR_SEQ、XSEQとする)、以下の条件に変換してチェックを行う方法もある。

$$XR_SEQ - N1 \leq XSEQ \leq XR_SEQ$$



N1: 重複メッセージと見なす通番値

図50 受信通番の有効範囲

D.3.2 1対1通信での処理

D.3.2.1 送信側ノードでの管理方法

< 割り当て >

- 送信通番のバージョンを管理するため、通番バージョン番号(V_SEQ)を、ノードに1つ設ける。
- 送信通番(S_SEQ)については、ノード毎のコネクション単位で管理する。
- 通番バージョン番号にセットする時刻は、秒単位とする。

< 初期化時の処理 >

システム立ち上げによる上記送信通番を初期化する場合は、初期化時刻を通番バージョン番号V_SEQにセットした後、S_SEQ値を1にする。

< 送信時の処理 >

メッセージ送信時、V_SEQをヘッダ内V_SEQ(バージョン番号)にセットし、S_SEQをヘッダ内SEQ(送信通番)にセットする。この時のS_SEQの値により次のように更新する。

◆ 通番S_SEQ = 0x7FFFFFFF時

S_SEQの値をカウントアップ(+1)する。

◆ 通番S_SEQ = 0x7FFFFFFF時

カウントアップせずにS_SEQを1に戻す。

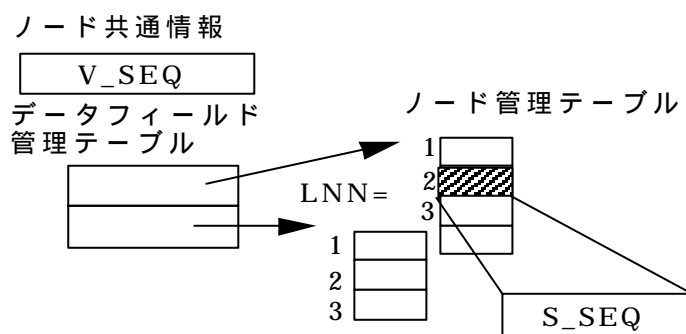


図51 1対1通信送信側ノードでの通番管理

D.3.2.2 受信側ノードでの管理方法

<前提条件>

- データフィールド毎かつ1対1接続通番管理テーブル(PTPSEQ)を持つ。
- 1対1通信メッセージ受信時、送信元ノードのDFNと優先レベルより、データフィールド管理テーブル経由で上記PTPSEQにアクセスする。
- 各PTPSEQは、LNNごとの配列になっており、各要素は最新受信通番(R_SEQ)と最新受信通番バージョン番号(R_V_SEQ)がセットされる。
- メッセージ送信元LNN毎に受信メッセージの最新通番(R_SEQ)と通番バージョン番号(R_V_SEQ)を保持する。

<初期化時の処理>

立ち上げ後、最初に受信したメッセージを無条件に受信するため、立ち上げ時の初期設定ではR_V_SEQ、R_SEQ共に"0"をセットする。

メッセージ受信時、ヘッダ内SAより送信元ノードのLNNとSEQ(送信通番)及びV_SEQ(通番バージョン番号)を読み込み、上記LNNと受信ポート番号からPTPSEQを検索し、R_V_SEQとR_SEQを求め、『D.3.1.3 マルチキャスト受信処理における通番チェック処理』と同じ処理を行う。

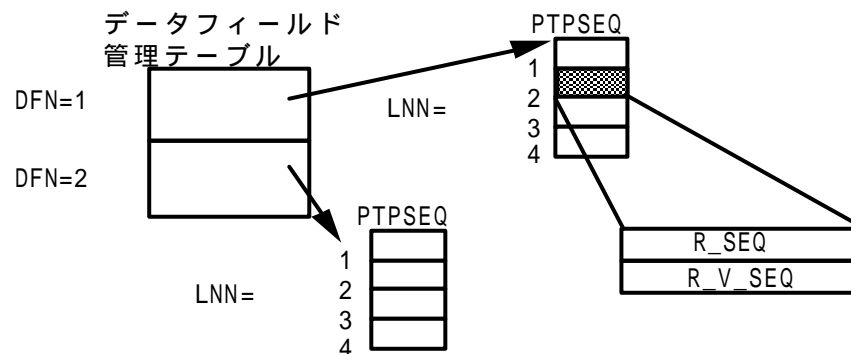


図52 1対1通信受信側ノードでの通番と通番バージョン番号管理

D.4 メッセージの分割・組み立て

D.4.1 メッセージの分割・組み立て処理

本仕様では1対1通信で扱えるユーザメッセージの最大サイズは、16Kバイトである。しかし、LAN上での最大転送サイズ(MTU)は、イーサネットの場合規格で1500バイトと定義されている。1対1通信では、IPヘッダの20バイト、TCPヘッダの20バイト、更にADPヘッダの64バイトを使用するため、実質的なデータ格納サイズは、(MTU-104)バイトである。

[注意]

TCPコネクション確立時のTCPセグメントサイズのネゴシエーションでTCPセグメントのユーザ最大サイズが1460バイトより小さいサイズになることもある。

メッセージ長が(MTU-104)よりも大きい場合、ひとつのメッセージを複数に分割し、ADPヘッダを付加してPDUを作成し、順にデータフィールドへ送信する。逆にメッセージ受信側では、複数のPDUを受信し、元のひとつのメッセージに再組み立てする。

図53 に1対1通信でのメッセージの分割、組み立ての概要を示す。

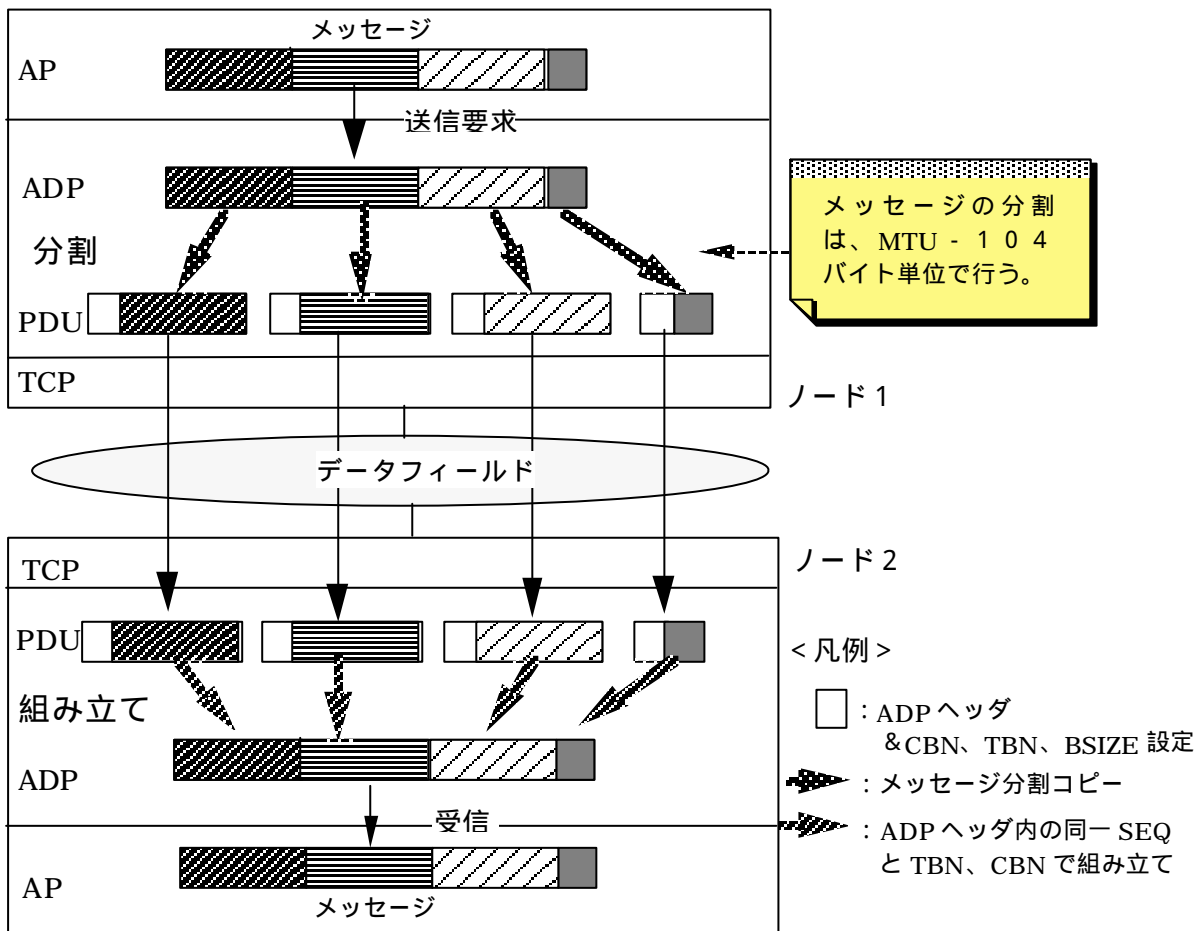


図53 メッセージの分割と再組み立て

D.4.2 分割、再組み立て用ヘッダ情報

次にヘッダ内のメッセージの分割、組み立てに使用する情報を示す。図53に示したメッセージ分割を例に 図54にこれら情報の設定例を示す。

CBN : カレントフラグメントPDU番号 (先頭PDU番号は1から)

TBN : トータルフラグメントPDU数

BSIZE : PDUサイズ

ML : メッセージサイズ + 64 (バイト)

SEQ : メッセージの通番






分割例	4分割時				分割なし
SEQ	100	100	100	100	101
CBN/TBN	1 / 4	2 / 4	3 / 4	4 / 4	1 / 1
BSIZE	1396	1396	1396	568	1024
ML	4564	4564	4564	4564	1024
パケット					

図54 メッセージ分割ヘッダ情報例

D.4.3 メッセージ分割アルゴリズム

メッセージ分割のアルゴリズムを次に示す。

```

UML = ユーザメッセージ長
if ( UML > 16384 )
エラー処理: 終了;
UBS = MTU - 104
TBN = (UML+UBS - 1) / UBS
if ( UBS < UML ) {
  for (CBN = 1, CBN = TBN, CBN++) {
    if (CBN = TBN )
      HD.BSIZE = UML - UBS*(CBN - 1)+64
    else
      HD.BSIZE = UBS+64
    HD.ML = UML+64
    HD.CBN = CBN、 HD.TBN = TBN
  }
}
else{
  HD.ML = UML+64、 HD.BSIZE = UML+64
  HD.CBN = 1、 HD.TBN = 1
}

```

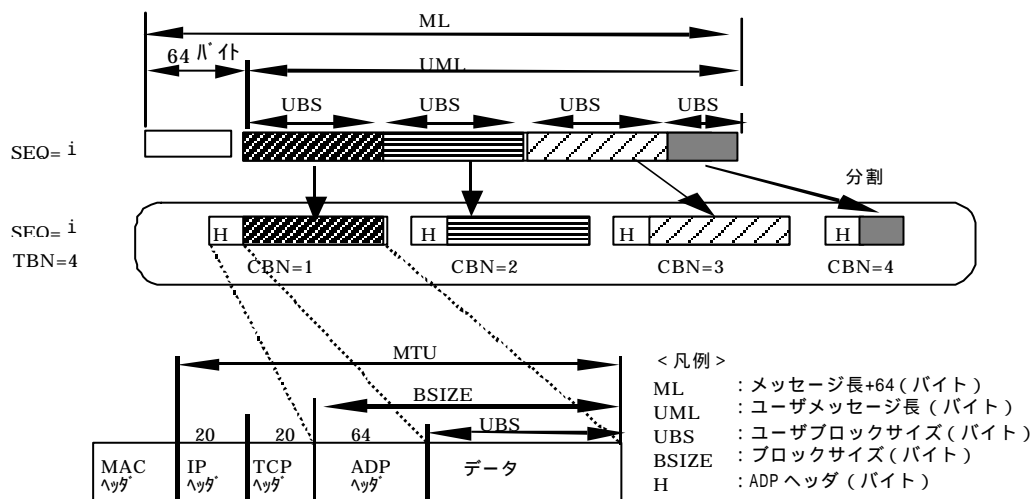


図55 メッセージからパケットへの分割

D.4.4 メッセージの組み立てアルゴリズム

メッセージの組み立てを行うためには、次のリソースの確保と管理が必要である。組み立てアルゴリズムはこれらリソース管理を前提にしている。

<前提リソース>

- メッセージ受信バッファ

分割されたメッセージのフラグメントPDUが全て受信できるまでの間保管するためのバッファである。

- フラグメントブロックビットマップ（順番チェック用）

分割されたメッセージのどのフラグメントパケットを受信済みかをチェックするためのビットマップである。

- 監視タイマ

メッセージの組み立てを開始してから永久に受信フラグメントPDUを保持するとバッファが満杯になるため分割パケットの到着間隔時間をタイムアウト監視する。タイムアウト時、組み立て中メッセージを廃棄する必要がある。タイムアウト時間は、デフォルト15秒としチューニング可能とする必要がある。

次にメッセージを再度組み立てるアルゴリズムを示す。

```

バッファ識別子 受信パケットのDFN、TCD、送信元ノード番号、V_SEQ
if (受信パケットのバッファ識別子と同じ識別子を持つ組み立て中メッセージあり
    && それは受信PDUのSEQよりも古い(小さい) )
{
    組み立て中メッセージの破棄処理
}
else /** else1 **/
if (CBN==1 && CBN==TBN) /* 以下受信PDUの処理 */
{ 完結しているメッセージ(分割されていない) 受信処理 }
else /** else2 **/
if ( 新規分割メッセージか )
{
    メッセージ再組み立て待ちリスト(バッファ) 確保
    ビットマップ確保し受信フラグメントPDUを受信済みに
    バッファ確保しデータ保存
    監視タイマスタート
}
else{
    監視タイマリセット
    メッセージ組み立て待ちリストをバッファ識別子から求める
    ビットマップに受信フラグメントPDUを受信済みに
    バッファ確保しデータ保存
    if (CBN==TBN && 全ブロック受信済み?)
    {
        リスト中のデータ部をユーザ受信エリアにコピーできる状態へ。
        終わり
    }
    else{ 監視タイマ再スタートし次フラグメント待ち }
}
} /** else2 **/
} /** else1 **/

```

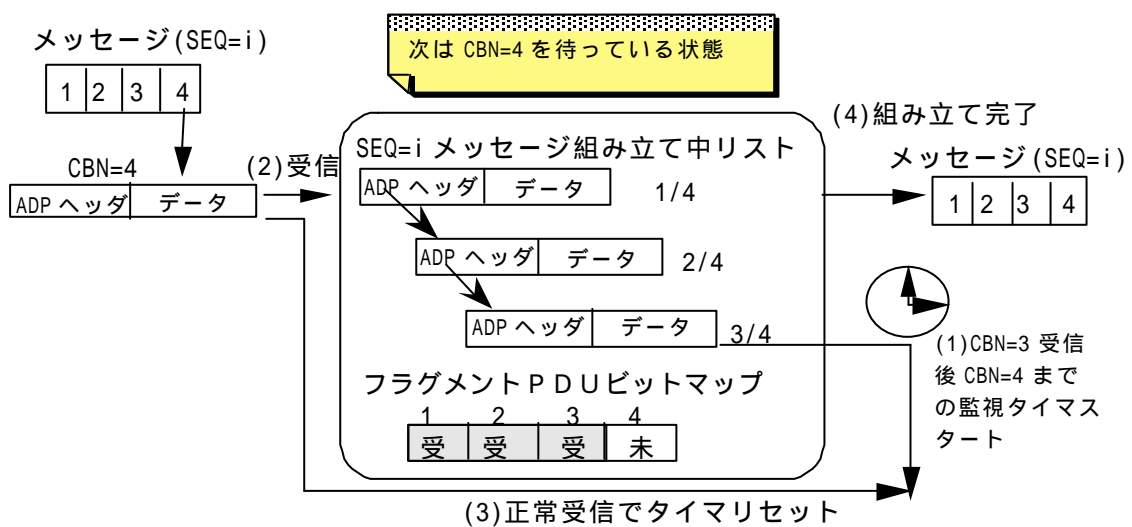


図56 メッセージの正常組み立て処理

次にメッセージを廃棄する条件と処理を示す。

< 廃棄の条件 >

次の要因が発生した場合である。

- 監視タイムアウト時
- 同一発行元ノードからの次通番のメッセージが到着した時
但し、次通番メッセージ到着で直ちに廃棄するか、あるいは、監視タイムアウト時間経過後に廃棄するかは実装にまかせる。

< 廃棄処理 >

- タイムアウトしたメッセージ組み立てリストのバッファを解放
- ビットマップリセット
- メッセージ廃棄したことをユーザへ通知
- 監視タイムアウトによる廃棄以外は、監視タイマリセット

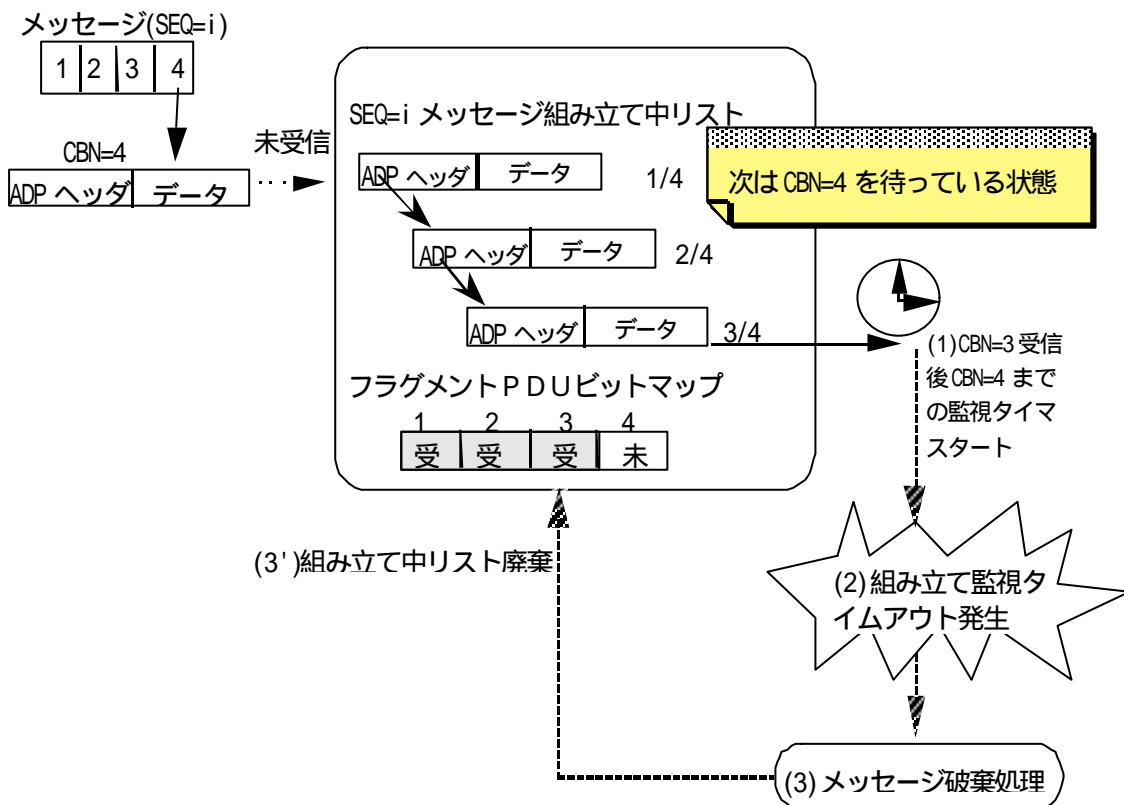
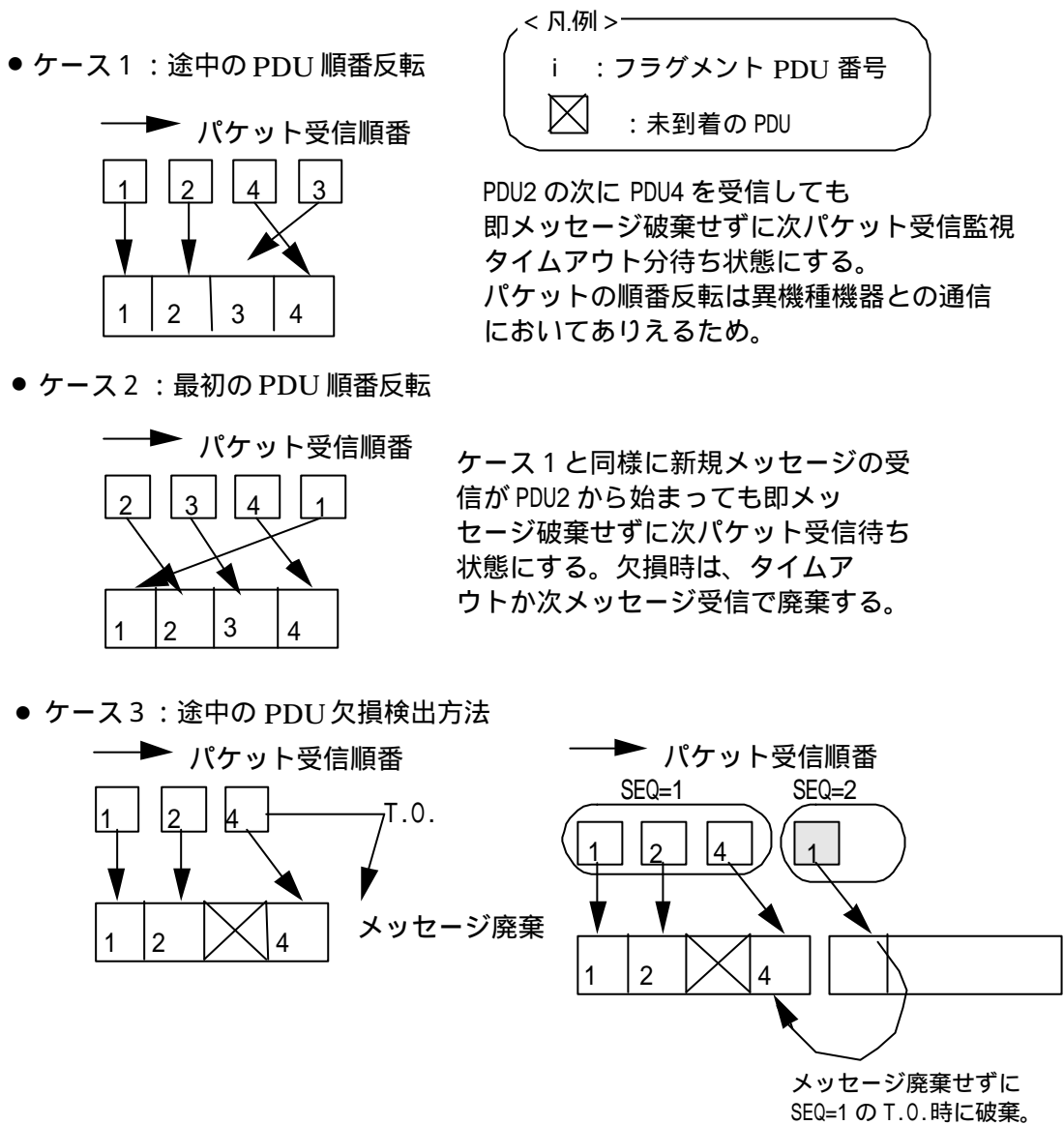


図57 メッセージの組み立て中タイムアウト処理

メッセージの組み立て処理を行う上で以下の図58に示す状態が発生することがある。これら異常状態への対処も組み込む必要がある。



< 凡例 >

i : フラグメント PDU 番号

☒ : 未到着の PDU

PDU2 の次に PDU4 を受信しても即メッセージ破棄せずに次パケット受信監視タイムアウト分待ち状態にする。パケットの順番反転は異機種機器との通信においてありえるため。

ケース 1 と同様に新規メッセージの受信が PDU2 から始まってても即メッセージ破棄せずに次パケット受信待ち状態にする。欠損時は、タイムアウトか次メッセージ受信で廃棄する。

受信メッセージの再組み立てで異常を検出したときは、受信バッファなしの場合を除けば、すぐにメッセージ廃棄するのではなく、メッセージ組み立て時間オーバー (T.O.) となったときに廃棄する。この時間は、チューニング可とする。

図58 メッセージ組み立て時の注意事項

D.5 ノード生死状態監視処理

データフィールドが1LANで構成される場合について、生存信号メッセージに基づく他ノード生死状態監視処理を以下に示す。

- 1) データフィールドの監視を開始した時点では、そのデータフィールドに属する全てのノードの状態は「死」の状態である。
- 2) あるノードから生存信号メッセージを受信した時点で、そのノードを「生」状態にあるまた、そのノードの生存信号タイムアウト監視時間 (al_tm_out) を記憶する。
- 3) 「生」状態にあるノードから、生存信号タイムアウト監視時間 (al_tm_out) ((2) で記憶した時間) を経過しても、生存信号メッセージが受信できない場合には、そのノードを「死」の状態であると判断する。
- 4) 生存信号メッセージの生存報告モード (al_mode) が、2 か 3 の場合には、そのノードを「死」の状態であると判断するとともに、その要因が、shutdown によるものか、メンテナンスによるものであるかを判断する。生存報告モード (al_mode) の判定は必須ではなく、必要に応じて実装するものとする。

[例] 生存信号タイムアウト時間(al_tm_out) 1 2 秒時の例

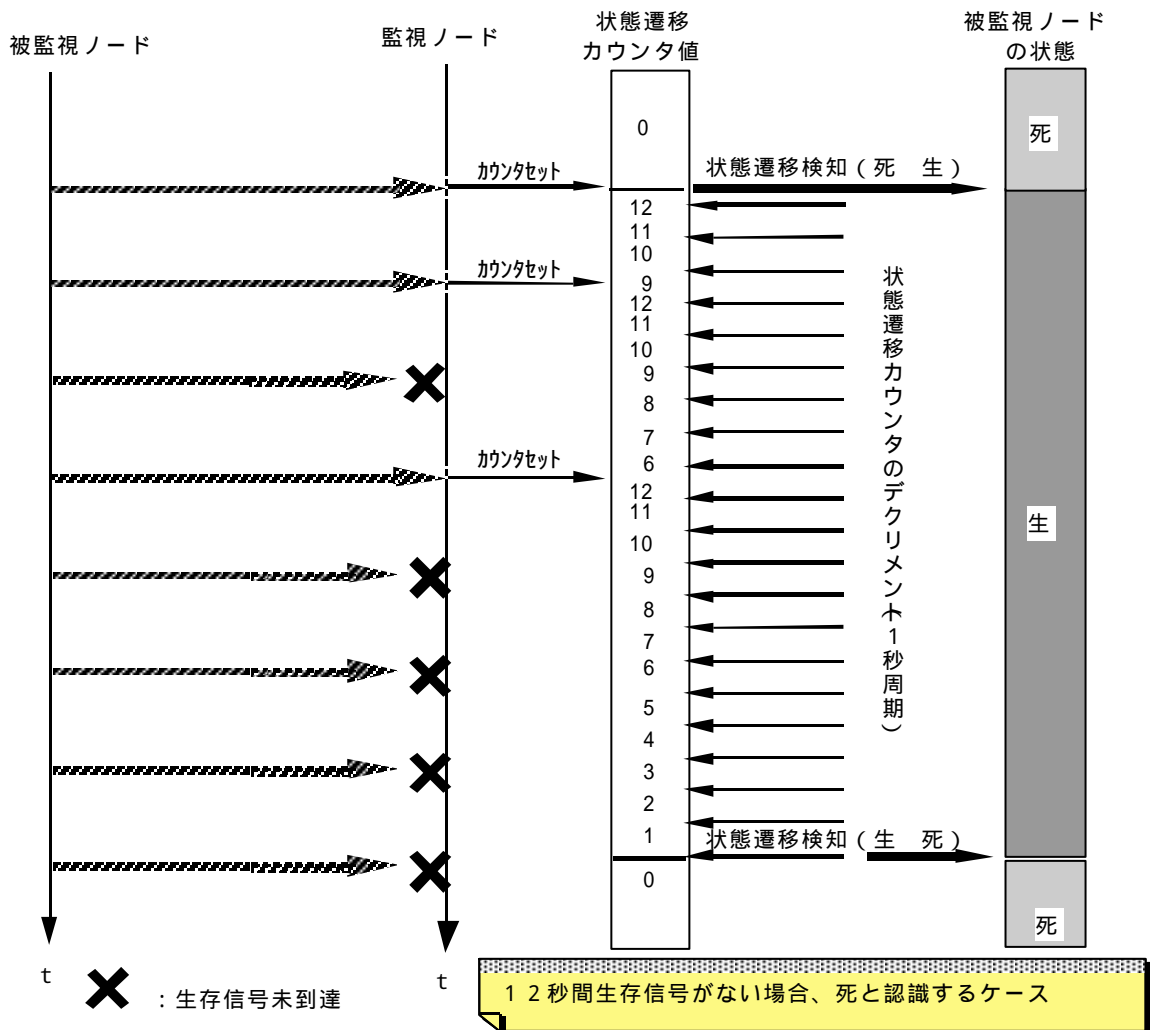


図59 生存信号監視シーケンス

D.6 障害情報送信方式

障害情報は、生存信号に付加した形で送出する。以下に、障害情報の作成方法を示します。送出する障害情報は、生存信号送出時点で作成する。

- (a) 生死を報告するモジュール数を `al_cnt_mod_alive` にセットする。
- (b) 生死を報告するモジュール数を 8 (1 バイトのビット数) で割って、4 で丸めた (long word 境界とするため) バイト数分の記憶領域 (`al_mod_alive`) を確保する。
- (c) 各モジュールの生死にしたがって、ビットにて生死を、確保した記憶領域 (`al_mod_alive`) にセットする。
- (d) 前回の障害情報送信 (生存信号送信) から発生した障害 (エラー) について、エラー番号とエラーの要因となっているモジュールをペアにしてリスト (`al_err_list`) に設定し、その数 `al_cnt_err_list` にセットする。障害 (エラー) 情報は、1 回以上送信する。なお、送達を確実にするために、1 度発生しただけのエラー (間欠エラー) についても、障害 (エラー) 情報として、生存信号タイムアウト時間 (`al_tm_out`) を生存信号送信間隔で割った数 (回数) だけ送信することを推奨する。
継続的に発生している障害については、毎回送信する。
- (e) エラー名称 (`al_err_name`: エラー番号の一意性を保つために付ける名前で、番号に対応したものではありません) を設定する。エラー名称は、監視ツールがエラー番号に対応するエラーメッセージを検索するファイルを特定するために使用する。(付属書 F.3 参照)
- (f) エラーコードは、2 バイトエリアを1 バイトずつ上位と下位に分けて管理し、上位バイトを概略、下位バイトを詳細コードとして対応させる。下記の通り体系化することを推奨する。

エラーコード 2 バイト
上位バイト: 概略コード
下位バイト: 詳細コード

概略コードの分類は、以下の通りとする。

0x01: イニシャルデータの誤り検出に伴うエラー
0x02: 異常ヘッダメッセージ受信検出に伴うエラー
0x03以上: 各ベンダーの自由
- (g) その他、障害情報として送信したい情報があれば、オプション情報 (`al_cnt_option`、`al_option`) としてセットする。

[注意]

以上のデータが、送信可能サイズ (MTU-28 バイト) を越える場合には、実装によって、`al_mode_alive` か `al_err_list` か `al_option` のいずれかの容量を減らす。たとえば、「エラーリストを複数回に分けて送信する」、「優先度の低いエラーは送信しない」などの対応が考えられる。

付属書 E . (参考) 実装例

E.1 バイトオーダーの問題

自律分散プロトコルでは、PDUのADPヘッダのバイトオーダーをビッグエンディアンと規定している。しかし、データ部（メッセージ）については、特に規定していない。データ部のバイトオーダーについては、ADPの上位層（AP）に一任している。

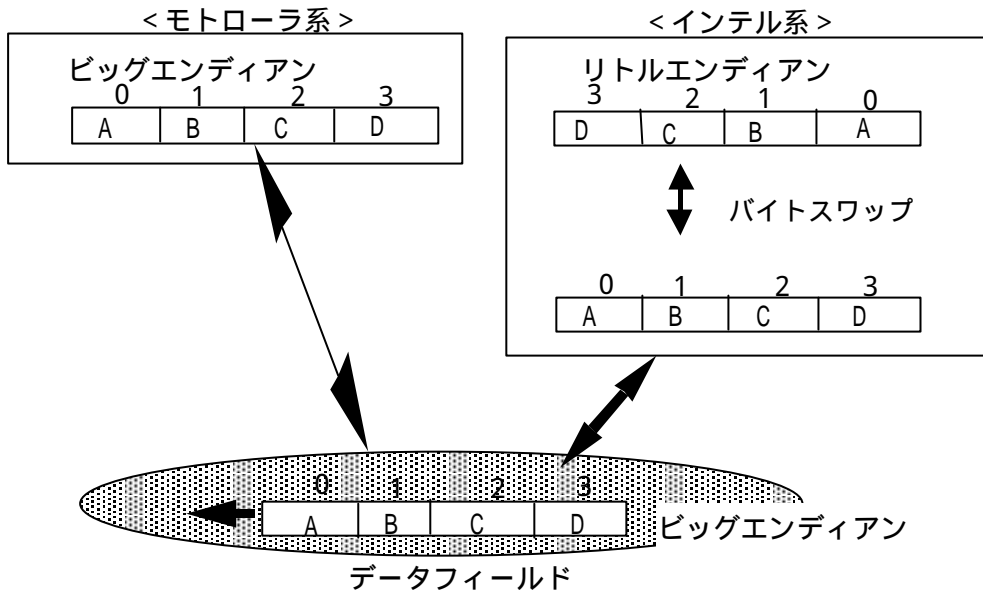


図60 ネットワークバイトオーダー

データ部のバイトオーダーについては、次の推奨方法を提案している。

◆ キャラクタ文字で転送

データ部を全てASCII文字等の文字列として転送する方法です。この方法によりAPにてバイトオーダーの違いを意識してバイトスワップを行う必要がなくなる。

◆ APにてバイトスワップ

リトルエンディアンのノード上のAPにてバイトオーダーをバイトスワップし、ノード上のバイトオーダーとネットワークバイトオーダーを変換する方法である。ただし、バイトオーダーがビッグエンディアンのノード上のAP、あるいはユーザデータが文字列の時はスワップ不要である。

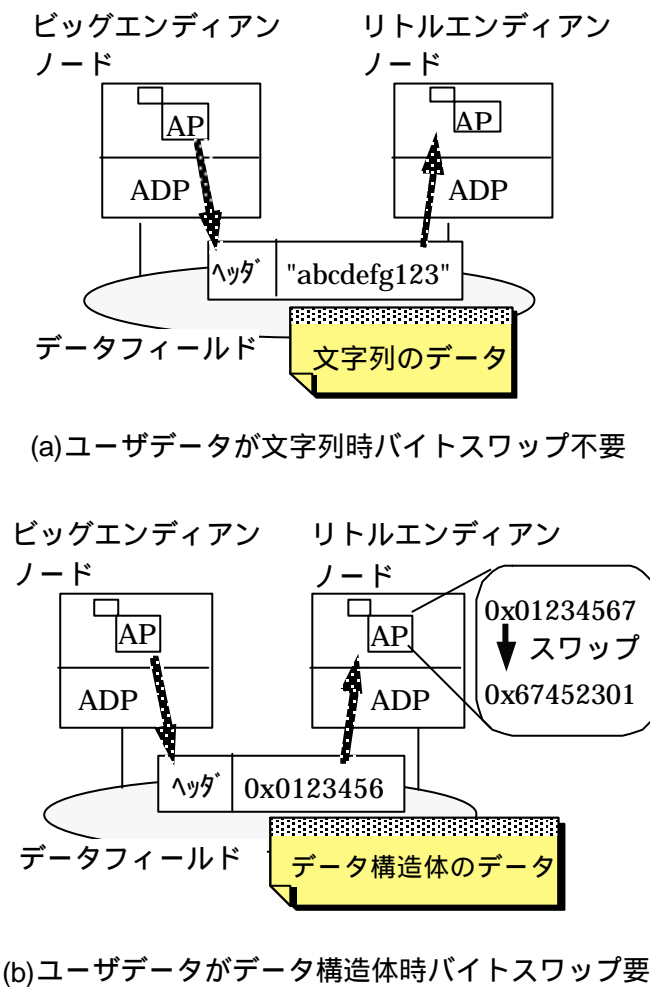


図61 バイトオーダーとバイトスワップ

E.2 アラインの問題

自律分散プロトコルでは、PDUのADPヘッダ部のヘッダ構造体のアラインをロングワードアラインにしており、各機器の言語のアラインの違いを吸収している。プロトコル実装時のヘッダについてはアラインの問題は発生しない。

しかし、データ部(メッセージ)については、バイトオーダ同様ADPの上位層(AP)に一任している。データ部のアラインについては、次の推奨方法を提案している。

◆ キャラクタ文字で転送

データ部を全てASCII文字等の文字列として転送する方法である。この方法によりAPにてアラインを意識する必要がなくなる。

◆ APにてロングワードアラインで統一

各ノード上のAP開発に使用する言語のアラインを意識しないために、データ構造体を全てロングワードアラインで定義する方法である。これによりワードアライン、ナチュラルアライン、ロングワードアラインの機器であっても問題なく通信できる。

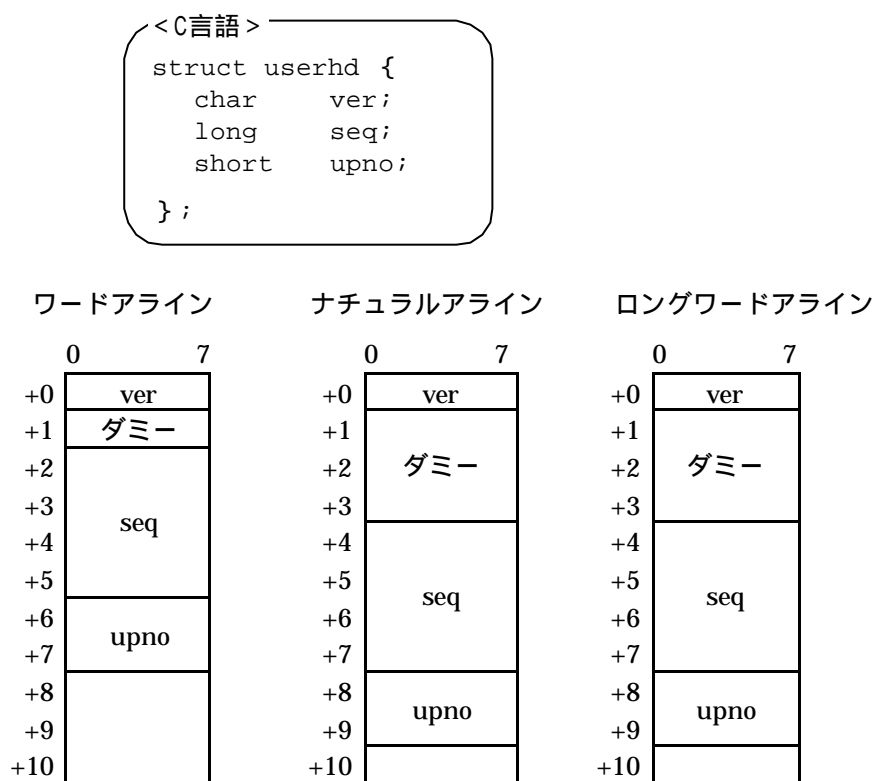


図62 アラインの違い

E.3 生存信号

E.3.1 条件

ネットワークアドレス : 128.0

ホストアドレス : 128.0.0.1

データフィールド : 1

自ノード番号 : 2

自ノード名 : "node2"

生存信号送出ポート : 600

生存信号送出周期 : 10 秒

E.3.2 生存信号メッセージの設定値

宛先の設定

ブロードキャストアドレス=128.0.255.255

宛先ポート=600

ADP ヘッダの設定

h_type = "NUXM"

ml = 128

sa = 0x00010002

da = 0x00010001

v_seq = 0

seq = 1

m_ctl = 0x80000000 # マルチキャスト

inq_id = 0

tcd = 60003

ver = 0

gtid = 0

mode = 0

pver = 1

pri = 1

cbn = 1

```

tbn = 1

bsize = m設定値(128)

fu1 = 0

    生存信号の設定

al_nd_name = "node2"

al_os_name = "HI_PC_win"    # 例 : H社製 P C (Windows系)

al_tm_out = 10 * 4

al_msgserno = 0

al_mode = 1

al_protocol = 4

al_chg_time = 0            # グリニッジ時刻のない機器時

al_ipaddr[0] = LANのIPアドレス

al_ipaddr[1] = 0

al_ver = 1

al_fu1 = 0

al_fu2 = 0

reserved = 0

    生存信号送信アルゴリズム

while ( ) {

    sendto ( socketno, 生存信号メッセージ格納アドレス、サイズ、宛先sockaddr )

    10 秒間delay

}

shutdown のとき生存信号の設定と送信例

al_mode = 2 # shutdown 予告

他情報は通常時とおなじ。

sendto ( socketno, 生存信号メッセージ格納アドレス、サイズ、宛先sockaddr )

```

E.4 障害情報の作成方法

あらかじめ各ノードでは、障害部位を特定するために障害部位（APあるいはハードウェア等）とモジュール番号を対応つける。そして、各障害部位で発生するエラー要因に対するエラー番号一覧を定義する。障害が発生時、モジュール番号を決定し、障害要因からエラー番号を決定する。モジュール番号付け、および障害要因とエラー番号の対応付けは、ADP実装者にて取り決める。

また、各障害には、オプション情報として、障害レジスタ情報等の詳細データを付加することができる。

図63の例は、ノード1内のAP毎にモジュール番号を対応させた場合の例である。モジュール番号3でメモリ不足エラーが発生し、障害要因からエラー番号を決定し、更に障害発生時の詳細データをオプション情報に格納し生存信号を送信している。

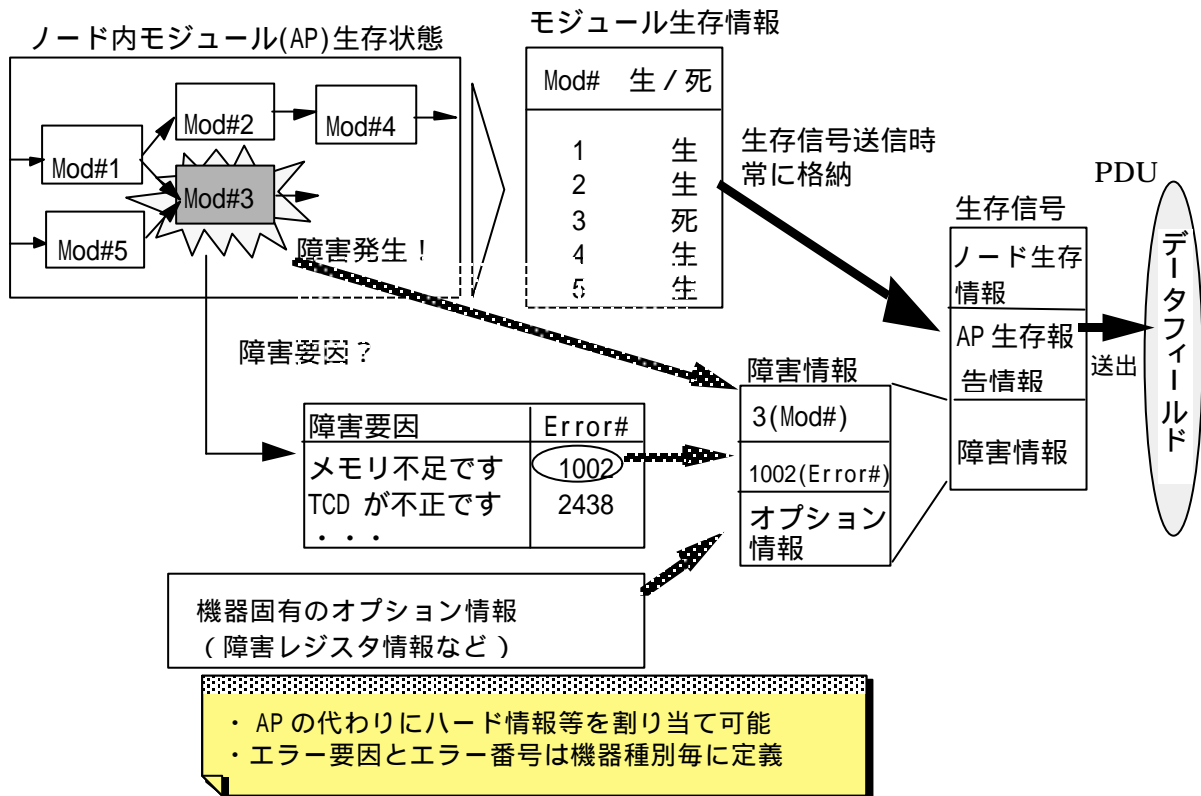


図63 障害発生時の障害情報の作成方法

付属書 F . (参考) 応用例

F.1 ノード状態の表示

生存信号を利用し、データフィールド内のノード状態を監視できる。

< UNIX環境での表示例 >

- 任意のデータフィールド内の全ノード状態をキャラクタ表示
- ノード状態変化をAPへオンライン通知可能

< Windows環境での表示例 >

- 任意のデータフィールド内の全ノードの生存状態を色分けしグラフィック表示
- ノード状態変化をDDE経由でAPへオンライン通知可能

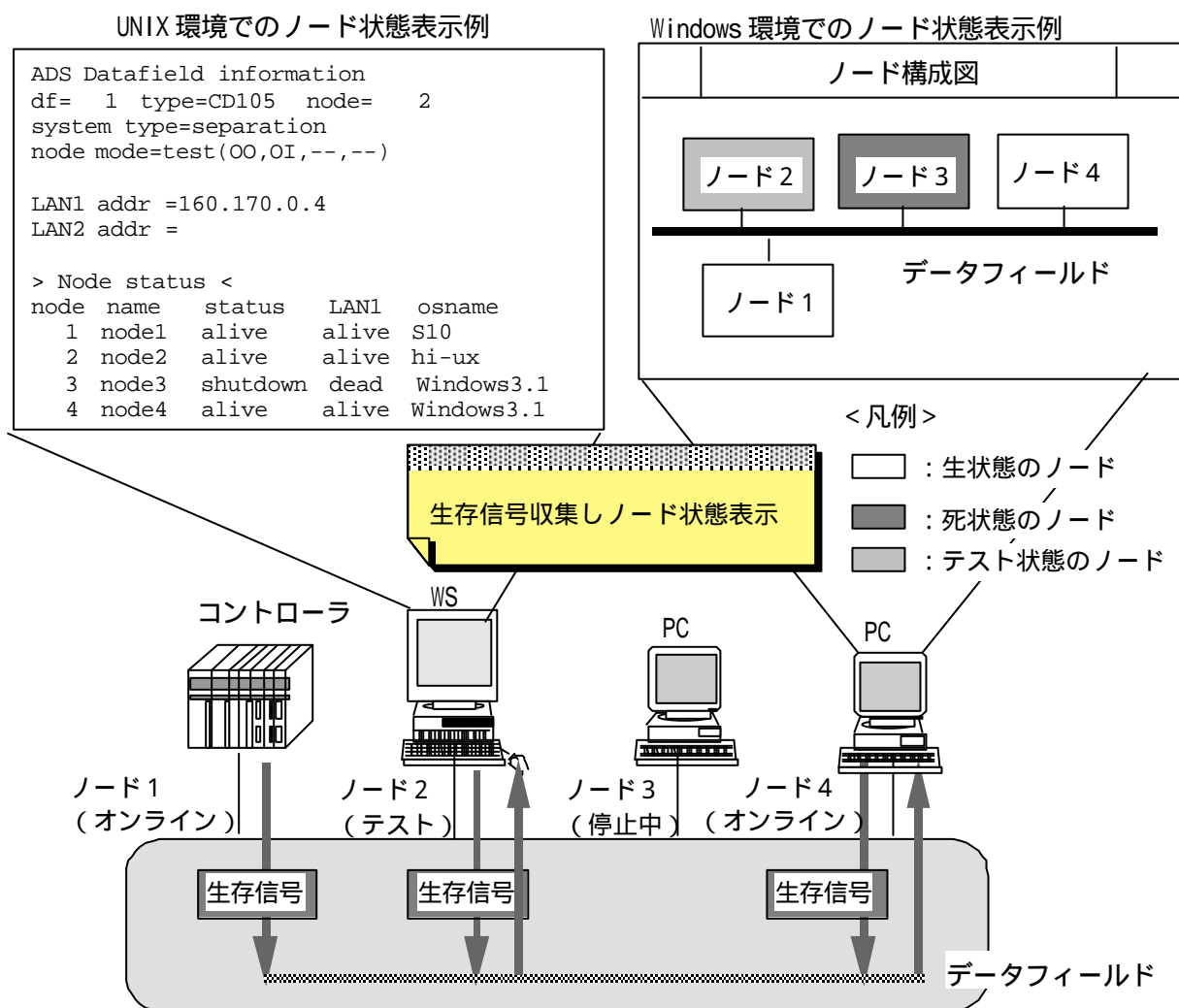


図64 ノードの生存監視

F.2 障害情報表示

生存信号に付加された障害情報は、障害情報を監視するノード上の監視APで障害監視が行える。障害情報を利用して例えば次のような表示が行える。

- モジュールの構成をグラフィック表示
- モジュールの生存状態を色分け表示
- ノード内発生エラーのリスト表示
- ノード内発生エラーのログおよび表示
- 機器固有のオプション情報の表示（バイナリ表示）

障害情報に対する表示例を以下に示す。この例では、ノード1のモジュール番号3のAPでメモリ不足が発生しAPが死状態に遷移した時の表示である。

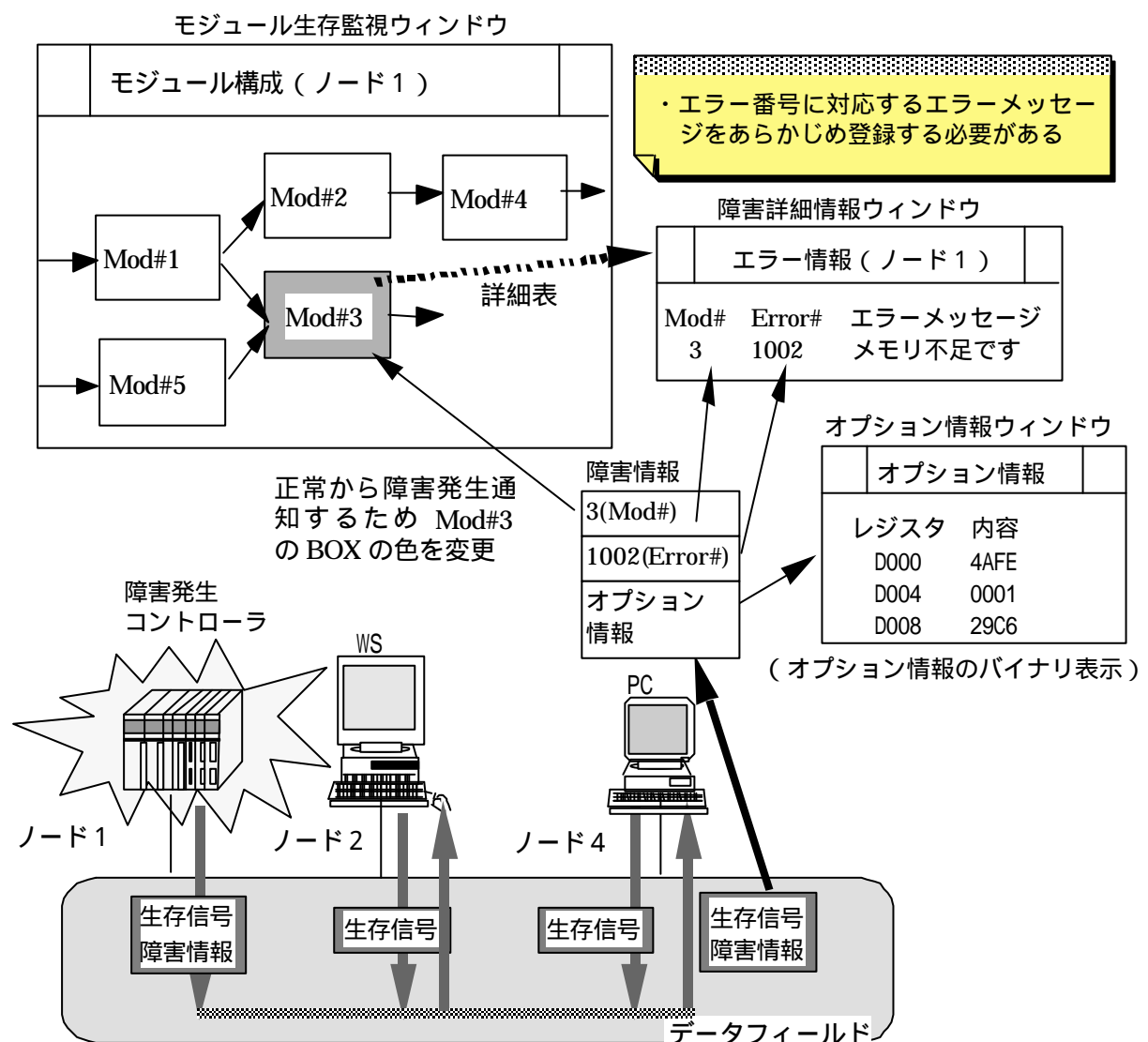


図65 障害情報の表示例

F.3 システム監視ツール用エラーメッセージ仕様

システム監視ツールは、受信した障害情報の中のエラー番号 (al_err_list) に対応したエラーメッセージを検索するために、以下に規定するエラーメッセージファイルを使用する。障害情報を送信する機器は、障害情報に対応するエラーメッセージファイルを作成し、システム監視ツール作成者に提供する必要がある。

システム監視ツールは、各ノードから送出された障害情報のエラー名称 (al_err_name) に対応するファイルがない場合、または、エラー番号 (al_err_list) に対応するエラーメッセージがない場合には、エラー番号とモジュール番号を障害情報として表示する。

このファイルの所在は、システム監視ツールによって規定されるディレクトリに格納できるようにする。

"ベンダ名_機器名" は、生存信号ヘッダの al_os_name に対応するものである。

No:	エラー番号	エラー種別	エラーシンボル
Msg:	エラーメッセージ (最大 1024 文字)		
Cause:	原因 (最大 1024 文字)		
Action:	対処法 (最大 1024 文字)		
		:	(上記の繰り返し)
		:	
		:	

・ASCII形式
・カラムの区切りはタブ

"No:" : 固定文字列。後に、エラー番号、エラー種別、エラーシンボルが続くことを表す。

エラー番号 : 10進数もしくは、0x で始まる16進数のエラー番号 (0~65535)

エラー種別 : "E" : エラー、"W" : 警告、"I" : 情報

エラーシンボル : エラー番号に対応付けたシンボル (省略可能)

"Msg:" : 固定文字列。後に続く文字列がエラーメッセージであることを表す。

"Cause:" : 固定文字列。後に続く文字列が原因の説明であることを表す。

"Action:" : 固定文字列。後に続く文字列が対処法の説明であることを表す。

図66 エラーメッセージファイルフォーマット

付属書 G . (参考) 自律分散システムの狙い

G.1 自律分散システムの狙い

自律分散システムアーキテクチャは、小規模分散システムから大規模分散システムまでをカバーし、システム内の制御サーバからワークステーション(WS)、パーソナルコンピュータ(PC)、更にコントローラ、各種制御機器などのコンポーネントを統合管理し、異機種、異文化の組み合わせで相乗効果を発揮させることができる。

自律分散システムアーキテクチャは、次の4つを主要な目的としており、多様化するユーザニーズに対応可能である。

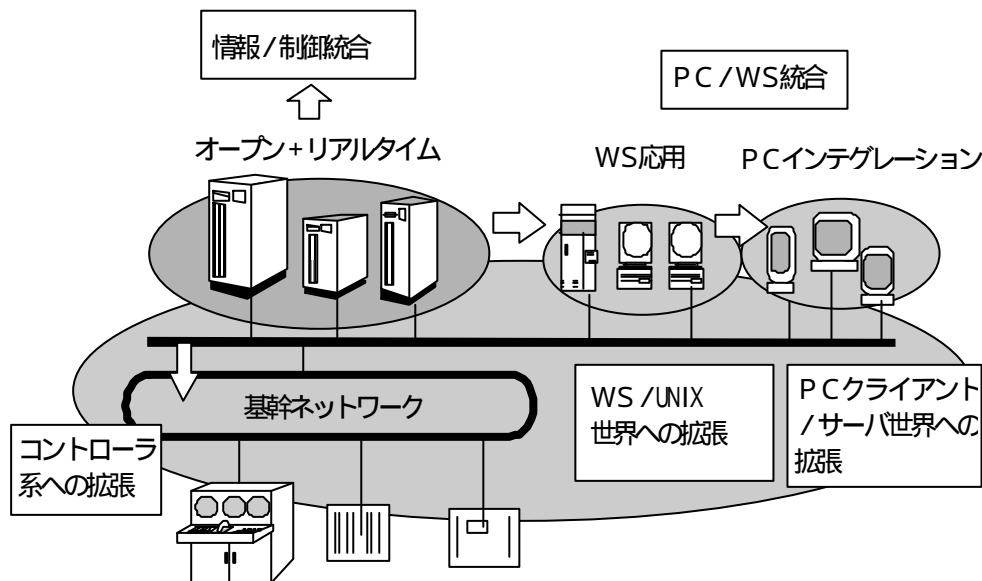
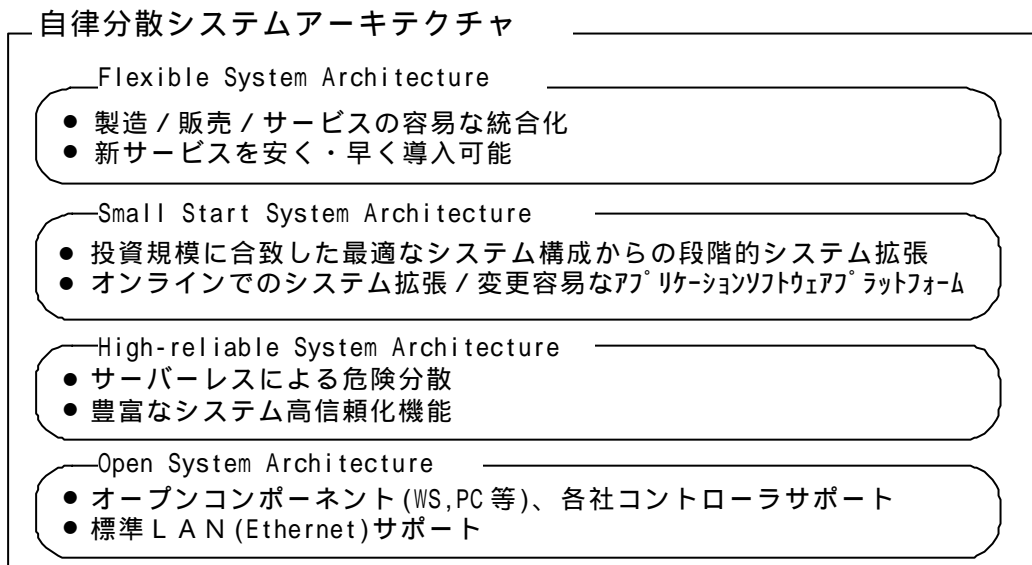


図67 自律分散システム

G.2 自律分散プロトコルの特長

本仕様で規定する自律分散プロトコルの特長を次に示す。

◆ 豊富な通信方式

特定の性格の情報、情報の質によって構成されるデータがながれる場を、データフィールドと呼ぶ。各ノードは、所属するデータフィールドにデータをマルチキャスト送信し、所属する他ノードは自律的にこのデータを受信することができる。システム全体を複数のデータフィールドに分割することにより、他サブシステムとの独立性が高められ、システムの拡張性/保守性を高めることができる。このマルチキャスト通信機能のほか、特定ノード間での1対1通信機能も提供しており、特定ノードへのダウンロードなど、ノード間の通信性質に合わせた通信機能を選択することができる。

◆ データフィールド管理機能

生存信号を使用することによりデータフィールドに接続する機器状態および機器内のプログラム、ハードウェア状態を自律的に収集することができ、機器、あるいは機器内のプログラムの構成管理、構成制御が行える。

◆ 障害監視機能

生存信号に障害情報を付加することで、データフィールドに接続する機器内のプログラム、ハードウェア状態に障害が発生した場合、障害部位の特定、障害要因、および障害の詳細情報をPC上で監視することができる。

◆ テスト支援機能

オンライン中のシステムでオンラインに影響を与えずにシステムテストを行えるようテスト支援する機能を備えている。

◆ 拡張性

既存のオンラインシステムを停止することなく、ノード増設、ユーザプログラム等のオンライン構築を行うことができ、システムの段階的な拡張が容易に行える。

付属書H . 連想配列の狙い

H.1 特徴

連想配列を用いたメッセージは、メッセージ中に、そのメッセージを構成するデータ項目名、データ型等の情報を含めて送信するもので、メッセージの自己記述性を高めることができる。

データの利用側はTCDとデータ項目名の組みで必要なデータを指定し、利用する。

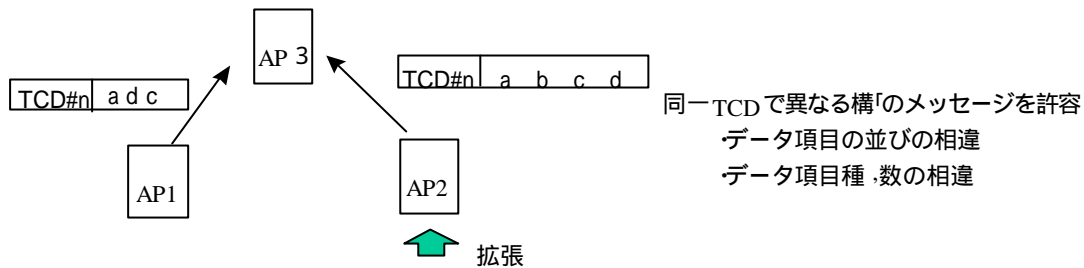


図68 連想配列におけるメッセージ概念

このため、以下の特長を得ることが出来る。

メッセージ解読容易性

- 1) メッセージの全体構造を知らなくとも、必要なデータ項目を取得可能である。
- 2) 柔軟性
- 3) 改造局所化：上例において、AP1,3が既設、AP3がデータ”b”を新たに必要とした場合に、AP1を改造することなくAP2及びAP3の改造のみで稼動可能である。
またはAP3を予めデータ”a”, ”b”, ”c”, ”d”に対応させて作成しておき、AP1, 2を段階的に（AP3を改造せずに）作成していける。
- 4) オンライン更新：メッセージデータ構造を改造する場合に、そのメッセージを用いている他のAPを止めることなく更新可能である。（データ項目の追加、並べ替えの場合）
- 5) TCD数削減

送信したいデータ項目のみを送信することができる。このため、1TCDの割り当てられたメッセージの構造を論理的に大きくすることが可能で、少ないTCDでデータを送信することができる。

H.2 本実装案での考え方

本委員会での審議において提案した実装案Rev.2は、前章の問題点を踏まえて、データ構造固定の従来型伝文と、データ構造可変の連想配列型伝文が同一メッセージにおいて共存できる形をとっていた。このためデータ部に固定部、拡張部といった区分が発生し、プロトコルが複雑であるとの意見があった。

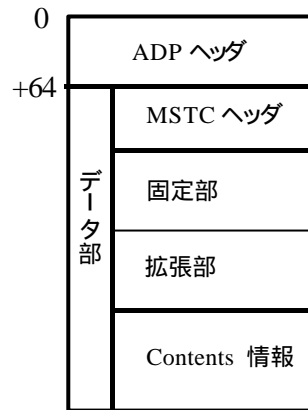


図69 連想配列実装案Rev.2での伝文構成

本実装案では、分かりやすさを優先し、固定構造を用いる従来型メッセージと、可変構造を用いる連想配列型メッセージを分けて考える。すなわち、あるメッセージがどちらの型であるかはMSTCヘッダにて規定され、後続するデータ部の解釈方法が決定する。

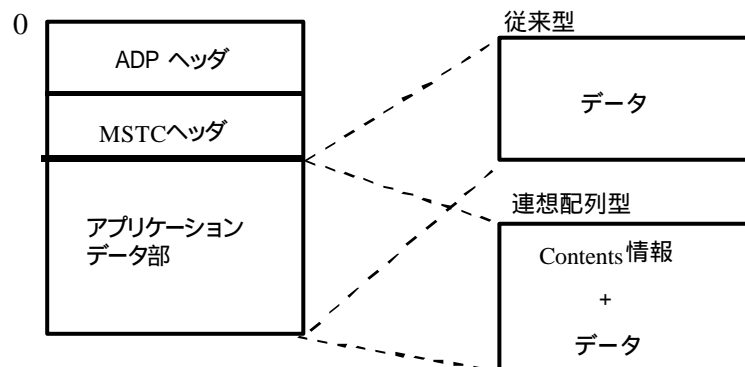


図70 従来型と連想配列の比較

MSTCヘッダを用いていない既存システムと、本プロトコルを用いたシステムを共存させる場合は、別々のDFを割り当てます（推奨）。ゲートウェイとなるAPがプロトコルの変換を行う。

DFを分けない場合は、既存メッセージのTCDと別のTCDを割り当て、TCD毎にどちらのプロトコルを用いるか決まるようにする。

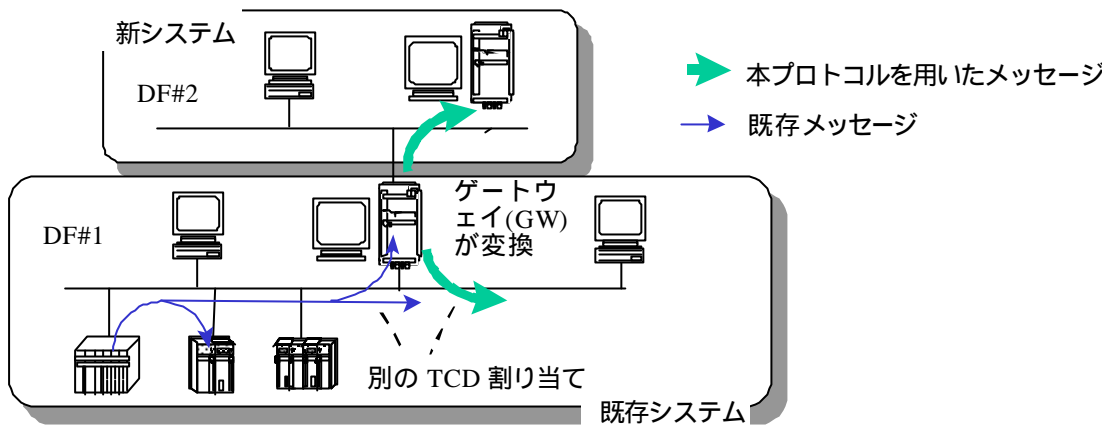


図71 DF分割例