

HITACHI

ソフトウェアマニュアル

オペレーション

NXACP For Windows[®]

S10VE

ソフトウェアマニュアル

オペレーション

NXACP For Windows®

SIOVE

この製品を輸出される場合には、『外国為替及び外国貿易法』の規制ならびに『米国輸出管理規則』など外国の輸出関連法規をご確認のうえ、必要な手続きをお取りください。
なお、ご不明な点がございましたら、当社担当営業にお問い合わせください。

2019年 9月 (第1版) SEJ-3-134 (A)

- このマニュアルの一部または全部を無断で転写したり複製したりすることは、固くお断りいたします。
- このマニュアルの内容を、改良のため予告なしに変更することがあります。

安全上のご注意

- システムの構築やプログラムの作成などは、このマニュアルの記載内容をよく読み、書かれている指示や注意を十分理解してから行ってください。誤操作により、システムが故障することがあります。
- このマニュアルは、必要なときすぐに参照できるよう、手近なところに保管してください。
- このマニュアルの記載内容について疑問点または不明点がございましたら、最寄りの弊社営業またはSEまでお知らせください。
- お客様の誤操作に起因する事故発生や損害については、弊社は責任を負いかねますのでご了承ください。
- 弊社提供ソフトウェアを改変して使用した場合に発生した事故や損害については、弊社は責任を負いかねますのでご了承ください。
- 弊社提供以外のソフトウェアを使用した場合の信頼性については、弊社は責任を負いかねますのでご了承ください。
- ファイルのバックアップ作業を日常業務に組み入れてください。ファイル装置の障害、ファイルアクセス中の停電、誤操作、その他何らかの原因によりファイルの内容を消失することがあります。このような事態に備え、計画的にファイルのバックアップを取っておいてください。
- 弊社製品が故障や誤動作したりプログラムに欠陥があった場合でも、使用されるシステムの安全が十分に確保されるよう、保護・安全回路は外部に設け、人身事故や重大な災害に対する安全対策が十分確保できるようなシステム設計としてください。
- 非常停止回路、インターロック回路などはPLCの外部で構成してください。PLCの故障により、機械の破損や事故の恐れがあります。
- 運転中のプログラム変更、強制出力、RUN、STOPなどは十分安全を確認してから行ってください。誤操作により、機械の破損や事故の恐れがあります。
- このマニュアルでは、安全上の注意事項のランクを潜在危険の重大度によって、「危険」、「警告」、「注意」、「通知」と区分しています。

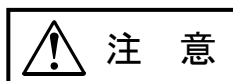
警告表示の定義



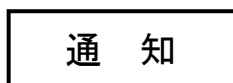
: この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こす危険の存在を示す。



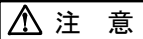

: この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こすおそれのある危険の存在を示す。



: この表示を無視して誤った取り扱いをすると、軽度の傷害または中程度の傷害を引き起こすおそれのある危険の存在を示す。



: この表示を無視して誤った取り扱いをすると、人身傷害とは関係のない損害を引き起こすおそれのある危険の存在を示す。

なお、 **注意**、 **通知** に記載した事項でも、状況によっては重大な結果に結びつく可能性があります。どれも重要な内容を記載していますので必ず守ってください。

「重大な傷害」、「軽度の傷害または中程度の傷害」、「人身傷害とは関係のない損害」について、具体的な内容を以下に示します。

重大な傷害

失明、けが、やけど（高温、低温）、感電傷害、骨折、中毒などで、後遺症が残るものおよび治療のために入院、長期の通院を要するもの

軽度の傷害または中程度の傷害

治療のために入院や長期の通院を必要としないけが、やけど、感電傷害など

人身傷害とは関係のない損害

周囲の財物の損傷、弊社製品の故障や破損、データの損失など、人身傷害以外の損害

安全上の注意事項は、安全性を確保するための原則に基づいた、弊社製品における各種対策を補完する重要なものです。弊社製品やマニュアルに表示されている安全上の注意事項は、十分に検討されたものですが、それでも、予測を超えた事態が起こることが考えられます。操作するときは指示に従うだけでなく、常に自分自身でも注意するようにしてください。また、弊社製品の安全な運転および保守のために、各種規格、基準に従って安全施策を確立してください。

来歴一覧表

改訂No.	来歴（改訂内容）	発行年月	備考
A	新規作成	2019.9	

このページは白紙です。

はじめに

このマニュアルは、S10VEシステムで協調自律分散システムを構築するための通信プラットフォームであるNXACPについて、その機能とマクロコールのリンケージ仕様を中心に解説するものです。

なお、このマニュアルは、このシステムのリアルタイムOSであるCPMS、Windows®やワークステーション（以降、WSと略します）で使用されるソケットの概念についての基礎知識を持っている読者を対象に記述されています。

<マニュアル構成>

第1編 概説

NXACPの特長、システム構成、基本的な用語説明、および機能概要について解説しています。

第2編 機能ガイド

NXACPのシステム構築、プログラム開発、テスト、運用で必要となる基本機能を次の8章に分類し解説しています。

第1章 マルチキャスト通信機能

第2章 データフィールド管理機能

第3章 二重化LAN制御機能

第4章 テスト機能

第5章 システム管理機能

第6章 運用管理機能

第7章 ネットワーク共有メモリ機能

第8章 システム構築機能

第3編 マクロ仕様

NXACPが提供するマクロコールの機能とリンケージ仕様を解説しています。

<関連マニュアル>

マニュアル名称	マニュアル番号
ユーザーズマニュアル S10VE総合編	SEJ-1-001
ユーザーズマニュアル S10VEへのリプレースマニュアル	SEJ-1-146
S10VE ソフトウェアマニュアル CPMS概説&マクロ仕様	SEJ-3-201
S10VE ソフトウェアマニュアル オペレーション RDP For Windows®	SEJ-3-133
NX Dlink ガイド	RS90-3-X050
NX Dlink リファレンス	RS90-3-X051

* マニュアル番号のXは改訂番号を示します。

<商標について>

Microsoft® Windows®は米国マイクロソフトコーポレーションの米国およびその他の国における登録商標です。

<記憶容量の計算値についての注意>

● 2ⁿ計算値の場合（メモリ容量・所要量、ファイル容量・所要量など）

1KB（キロバイト）＝1,024バイトの計算値です。

1MB（メガバイト）＝1,048,576バイトの計算値です。

1GB（ギガバイト）＝1,073,741,824バイトの計算値です。

1TB（テラバイト）＝1,099,511,627,776バイトの計算値です。

● 10ⁿ計算値の場合（ディスク容量など）

1KB（キロバイト）＝1,000バイトの計算値です。

1MB（メガバイト）＝1,000²バイトの計算値です。

1GB（ギガバイト）＝1,000³バイトの計算値です。

1TB（テラバイト）＝1,000⁴バイトの計算値です。

目次

第1編 概説

第1章 概要	1-2
1.1 NXACPとは	1-2
1.2 ハードウェア構成	1-3
1.3 ソフトウェア構成	1-4
第2章 用語の説明	1-5
2.1 基本用語の説明	1-5
第3章 機能概要	1-10
3.1 マルチキャスト通信機能	1-10
3.2 データフィールド管理機能	1-13
3.3 二重化LAN制御機能	1-15
3.4 テスト機能	1-16
3.5 システム管理機能	1-17
3.6 運用管理機能	1-19
3.7 ネットワーク共有メモリ機能	1-20
3.8 システム構築機能	1-22

第2編 機能ガイド

第1章 マルチキャスト通信機能	2-2
1.1 通信の特長	2-2
1.1.1 メッセージ通信の単位	2-2
1.1.2 メッセージの分割・組み立て	2-3
1.1.3 ユーザプログラム構造	2-4
1.1.4 メッセージ処理順序	2-8
1.2 メッセージ送信機能	2-10
1.2.1 ネットワーク送信	2-10
1.2.2 自ノード内送信	2-17
1.3 メッセージ受信機能	2-18
1.3.1 メッセージ受信	2-18
1.3.2 受信タイムアウト監視	2-22
1.3.3 マルチキャストグループの定義	2-23
1.4 リモートデータフィールド制御機能	2-25

1. 4. 1 システム接続形態	2-25
1. 4. 2 メッセージ送受信	2-27
1. 5 バッファ管理	2-29
第2章 データフィールド管理機能	2-30
2. 1 生存信号送信機能	2-30
2. 2 生存状態監視と状態変化通知機能	2-32
第3章 二重化LAN制御機能	2-35
3. 1 二重化LANでのメッセージ送受信	2-35
3. 2 二重化LANでの生存信号送受信	2-38
第4章 テスト機能	2-39
4. 1 メッセージ入出力制御	2-39
4. 2 テスト形態	2-41
4. 3 リモートデータフィールドとモード	2-43
第5章 システム管理機能	2-45
5. 1 障害通知機能	2-45
5. 2 ユーザタスク管理機能	2-51
第6章 運用管理機能	2-53
6. 1 NXACPとデータフィールドの運用管理	2-54
6. 1. 1 データフィールド起動	2-55
6. 1. 2 データフィールド停止	2-57
6. 1. 3 モードの設定と変更	2-59
6. 1. 4 構築情報の変更	2-60
第7章 ネットワーク共有メモリ機能	2-61
7. 1 用語説明	2-61
7. 2 転写メモリの仕様	2-63
7. 3 構築時の留意事項	2-69
7. 4 マクロ概要	2-70
第8章 システム構築機能	2-72
8. 1 NXACP本体のローディング	2-74
8. 2 システム構築とローディング	2-77
8. 2. 1 初期設定	2-77

8. 2. 2	構築情報の設定	2-78
8. 2. 3	構築情報のコンパイル	2-85
8. 2. 4	構築情報のローディング	2-86
8. 3	容量見積り	2-87

第3編 マクロ仕様

第1章	総説	3-2
-----	----	-----

1. 1	マクロの種類と一覧	3-2
------	-----------	-----

第2章	マルチキャスト通信マクロ	3-3
-----	--------------	-----

2. 1	nx_put	3-3
------	--------	-----

2. 2	nx_get	3-5
------	--------	-----

第3章	運用マクロ	3-7
-----	-------	-----

3. 1	nx_init	3-7
------	---------	-----

3. 2	nx_dfup	3-8
------	---------	-----

3. 3	nx_dfdwn	3-11
------	----------	------

3. 4	nx_quit	3-12
------	---------	------

第4章	共有メモリマクロ	3-13
-----	----------	------

4. 1	nx_init_tm	3-13
------	------------	------

4. 2	nx_ctl_tm	3-17
------	-----------	------

4. 3	nx_get_tm	3-19
------	-----------	------

4. 4	nx_write_tm	3-20
------	-------------	------

4. 5	nx_read_tm	3-22
------	------------	------

付録

付録A	リターンコード詳細	A-2
-----	-----------	-----

付録B	ログフォーマット	A-9
-----	----------	-----

付録C	ノード状態変化通知フォーマット	A-15
-----	-----------------	------

付録D	DHPレコード一覧	A-16
-----	-----------	------

付録E	制御トレースについて	A-17
-----	------------	------

付録F	メッセージヘッダフォーマット	A-18
付録G	生存信号ヘッダフォーマット	A-19

図目次

図 1-1	ネットワーク最小構成	1-3
図 1-2	ネットワーク最大構成	1-3
図 1-3	ソフトウェア構成	1-4
図 1-4	NXACP要素の説明	1-5
図 1-5	データフィールド	1-6
図 1-6	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	1-6
図 1-7	メッセージ	1-7
図 1-8	トランザクションコード	1-7
図 1-9	論理ノード	1-8
図 1-10	マルチキャストノード1グループ	1-8
図 1-11	ポート	1-9
図 1-12	マルチキャスト通信	1-10
図 1-13	マルチキャストグループ	1-10
図 1-14	メッセージの分割・再生	1-11
図 1-15	メッセージの切り出し受信	1-11
図 1-16	メッセージのフィルタリング	1-12
図 1-17	生存信号	1-13
図 1-18	ノード状態管理	1-13
図 1-19	状態監視ノード (1)	1-14
図 1-20	状態監視ノード (2)	1-14
図 1-21	LANの二重化	1-15
図 1-22	二重化LAN通信方式	1-15
図 1-23	テスト機能	1-16
図 1-24	障害通知	1-17
図 1-25	トレース	1-18
図 1-26	UPタスク管理	1-18
図 1-27	NXACP運用管理	1-19
図 1-28	ネットワーク共有メモリ	1-20
図 1-29	共有メモリの特長	1-21
図 1-30	システム構築	1-22
図 1-31	NXACPの構築環境	1-22
図 2-1	通信単位	2-2
図 2-2	nx_putマクロ	2-4
図 2-3	障害情報の検知 (送信)	2-5
図 2-4	nx_getマクロ	2-6
図 2-5	障害情報の検知 (受信)	2-7

図 2-6	処理順序	2-8
図 2-7	優先レベル使用時の制限事項	2-9
図 2-8	データフィールド番号と送信範囲	2-10
図 2-9	データフィールド番号とネットワークセグメント	2-10
図 2-10	マルチキャストと選択受信	2-11
図 2-11	リモートデータフィールド (1)	2-12
図 2-12	リモートデータフィールド (2)	2-12
図 2-13	リモートデータフィールド (3)	2-13
図 2-14	特定ノード宛送信 (1)	2-15
図 2-15	特定ノード宛送信 (2)	2-16
図 2-16	自ノード内通信	2-17
図 2-17	<code>nx_get()</code> の特長	2-18
図 2-18	受信処理の流れ	2-19
図 2-19	共用受信	2-20
図 2-20	切り出し受信	2-20
図 2-21	切り出し受信と使用メモリ	2-20
図 2-22	宛先送信マルチキャストグループと受信マルチキャストグループ	2-24
図 2-23	データフィールドタイプ	2-25
図 2-24	リモートデータフィールド (1)	2-26
図 2-25	リモートデータフィールド (2)	2-26
図 2-26	リモートデータフィールドのメッセージ送受信	2-28
図 2-27	生存信号	2-30
図 2-28	変化通知	2-32
図 2-29	変化通知のタイムシーケンス (タイムアウト)	2-33
図 2-30	変化通知のタイムシーケンス (SHUTDOWN予告)	2-33
図 2-31	二重化LANでのメッセージ送受信	2-35
図 2-32	二重化LANとリモートデータフィールド (1)	2-36
図 2-33	二重化LANとリモートデータフィールド (2)	2-36
図 2-34	二重化LANとリモートデータフィールド (3)	2-37
図 2-35	二重化LANとリモートデータフィールド (4)	2-37
図 2-36	二重化LANと生存信号	2-38
図 2-37	メッセージ入出力制御	2-40
図 2-38	テスト形態 (1)	2-41
図 2-39	テスト形態 (2)	2-41
図 2-40	テスト形態 (3)	2-42
図 2-41	リモートデータフィールドとモード (1)	2-43
図 2-42	リモートデータフィールドとモード (2)	2-44
図 2-43	リモートデータフィールドとモード (3)	2-44

図 2-44	構成例	2-47
図 2-45	バッファステータス出力タイミング	2-48
図 2-46	メッセージ到着時のタスク状態チェック	2-51
図 2-47	ABORT後処理	2-52
図 2-48	NXACPの運用	2-53
図 2-49	NXACPとデータフィールドの運用	2-53
図 2-50	各マクロと運用	2-54
図 2-51	データフィールドの初期化順序	2-55
図 2-52	リモートデータフィールドへの影響 (1)	2-58
図 2-53	リモートデータフィールドへの影響 (2)	2-58
図 2-54	モード変更手順	2-59
図 2-55	ソフト転写メモリ	2-61
図 2-56	転写メモリ識別子 (TMID)	2-62
図 2-57	書き込み/読み出しエリア	2-62
図 2-58	転写メモリ認識とマルチキャストグループ	2-64
図 2-59	ソフト転写時の排他制御	2-65
図 2-60	ソフト転写時の重複チェック	2-66
図 2-61	周期送信型ソフト転写の処理	2-67
図 2-62	分割送信型ソフト転写の処理	2-67
図 2-63	転写メモリの状態遷移	2-70
図 2-64	転写メモリとデータフィールド状態	2-71
図 2-65	リソースの配置	2-72
図 2-66	NXACPの構築手順	2-73
図 2-67	構築環境	2-77

表目次

表 1-1	カテゴリ A通信機能一覧	1-2
表 1-2	通信仕様一覧	1-2
表 2-1	バッファサイズ	2-3
表 2-2	マルチキャストグループとポート	2-11
表 2-3	トランザクションコードとユーザタスク対応	2-14
表 2-4	マルチキャストグループとノード対応	2-15
表 2-5	トランザクションコードとノード対応	2-16
表 2-6	トランザクションコードとユーザタスク対応	2-19
表 2-7	マルチキャストグループ番号と受信ポート番号	2-23
表 2-8	マルチキャストグループ番号と送受信ポート番号	2-23
表 2-9	データフィールドタイプとバッファ定義	2-29
表 2-10	生存信号タイプ	2-31
表 2-11	生存信号付加情報	2-31
表 2-12	ノードモードとメッセージモード	2-39
表 2-13	モードとポート番号	2-40
表 2-14	EAS通知イベント一覧	2-45
表 2-15	障害監視例	2-47
表 2-16	データフィールド起動処理	2-55
表 2-17	データフィールド停止処理	2-57
表 2-18	モード指定の有効範囲	2-59
表 2-19	転写種別単位の属性	2-63
表 2-20	転写サポート機種一覧	2-68
表 2-21	転写マクロ一覧	2-70
表 2-22	リソース割り付け	2-75
表 2-23	構築ファイル一覧	2-78
表 2-24	容量算出シート	2-87
表 3-1	マクロ一覧	3-2
表 3-2	パラメータ一覧	3-16

第1編 概説

第1章 概要

1. 1 NXACPとは

NXACPとは、プログラマブルコントローラ対応自律分散オンラインパッケージの総称で、NXACP/S10VEは、S10VEシステムを対象とした通信プラットフォームです。また、WSやパソコン（以降、PCと略します）上の自律分散オンラインパッケージであるNX Dlinkとともに協調して動作し、自律分散ネットワーク環境を提供することを目的としたソフトウェアです。

NXACP/S10VEは、「オープン自律分散インタフェース技術説明書 V1.1」に示す『カテゴリA』の下記機能をサポートしています。

表 1 - 1 カテゴリA通信機能一覧

第一次機能			第二次機能		
クラス名	機能名		クラス名	機能名	
A-Base-1	マルチキャスト通信機能	○	A-Opt-1-a	二重化LAN制御マルチキャスト通信機能	○
A-Base-2	生存信号送出機能	○	A-Opt-2-a	障害情報送出機能	○
A-Opt-3	1対1通信機能（最大16KB）	×			
A-Opt-4	ネットワーク経由構築機能	×			

○：サポート、×：未サポート

【補足】 協調自律分散プロトコル仕様に持つ優先制御は未サポートです。

NXACP/S10VEでは、優先レベル（PRI）を0固定でメッセージ送信します。

また、メッセージ受信時も優先レベルに従った優先処理はせず、到着順に処理します。

NXACP/S10VEの主な通信仕様を以下に示します。

表 1 - 2 通信仕様一覧

項目	仕様
ネットワーク	イーサネット
通信プロトコル	UDP/IP
メッセージサイズ	最大16384バイト
ローカルデータフィールド登録数	最大7DF（自ノード内通信含む）
リモートデータフィールド登録数	最大8DF/1ローカルデータフィールド
データフィールド登録総数	最大16DF
送信マルチキャストグループ登録数	最大128グループ
受信マルチキャストグループ登録数	最大32グループ（1DFにつき）
出力トランザクションコード数	最大8TCD/1タスク（1DFにつき）
入力トランザクションコード数	最大8TCD/1タスク（1DFにつき）
同一トランザクションコード入力タスク数	最大8タスク/1TCD（1DFにつき）

【補足】 各情報の指定有効範囲については「第2編 第8章 システム構築機能」を参照してください。

1. 2 ハードウェア構成

このシステムの機能を利用するために必要なハードウェア構成を図1-1、図1-2に示します。

● 最小構成

内蔵Ether経由イーサネットにPOC、PADTまたはPCと接続した構成が最小構成となります。

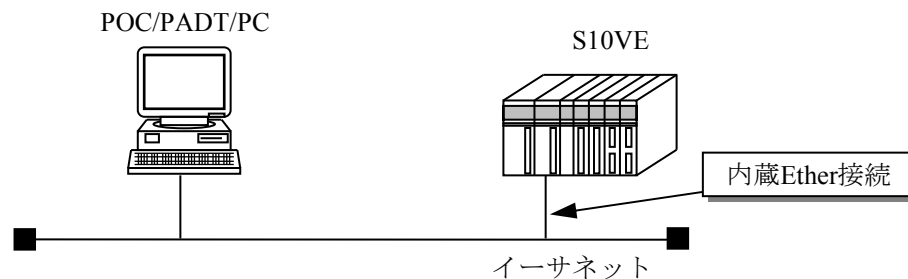


図1-1 ネットワーク最小構成

● 最大構成

最大構成は、2本のネットワークを使用した場合（内蔵Etherの二重化）となります。

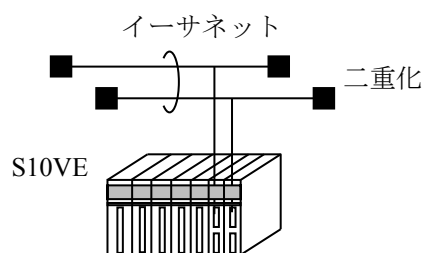


図1-2 ネットワーク最大構成

ルーティング機能を使用して、ルータやゲートウェイを経由したネットワークとも送受信できます。ただし、そのネットワークセグメント数は、1つの直接接続されたネットワークセグメントに対し最大8セグメントで、1つのS10VEから送受信できるネットワークセグメントは、直接接続されたネットワークとルータなどを経由したネットワークをすべて合わせて16セグメントまでです（自ノード内通信用のデータフィールドを定義した場合は、15セグメントまでです）。

1.3 ソフトウェア構成

このシステムの位置付け、ソフトウェア構成を図1-3に示します。また、前提とするソフトウェアは以下のとおりです。

- ・ CPMS/S10VE (リアルタイムモニタシステム)
- ・ RCTLNET/S10VE (イーサネットドライバ)
- ・ RPDP/S10VE (リアルタイムシステム開発パッケージ)
- ・ BASE SYSTEM/S10VE (基本システム)

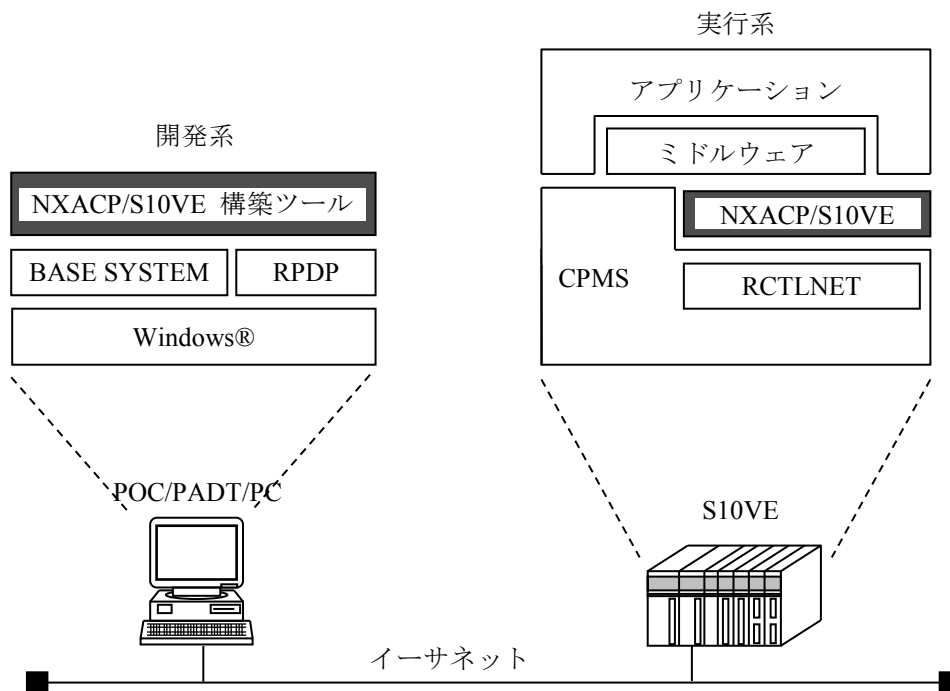


図1-3 ソフトウェア構成

第2章 用語の説明

2.1 基本用語の説明

NXACPは、図1-4に示す要素群で構成します。

NXACPでは、従来のネットワークパッケージとは異なり、ネットワーク構成に依存しない抽象的な用語を使用します。

この節では、図1-4の各構成要素について説明します。図1-4に示す()内の数字は、説明番号に対応しています。

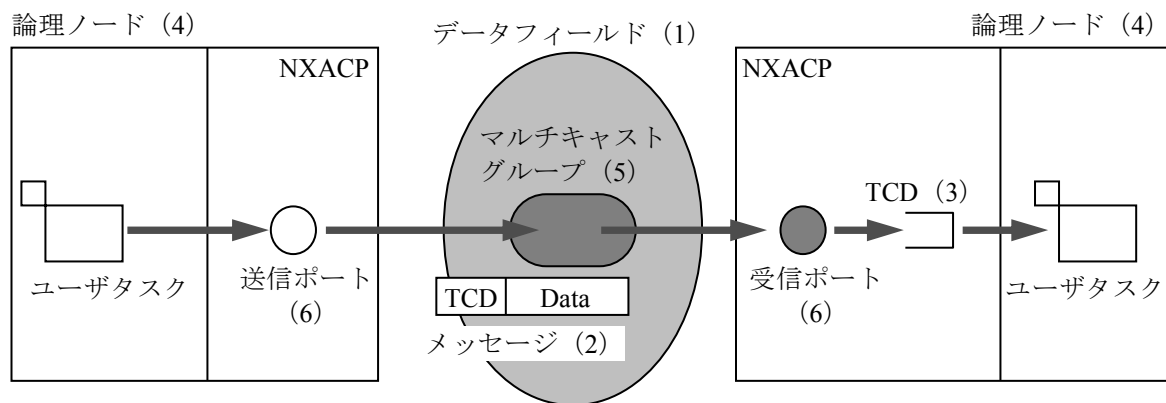


図1-4 NXACP要素の説明

(1) データフィールド

特定の属性情報によって構成されるメッセージの流れる場をデータフィールド (DF) と呼びます。NXACPでは、データフィールドは1つのLANセグメントに対応し、複数の機器が属します。つまり、データフィールドは、送受信するメッセージの伝送範囲を決めるために使用します。分散システム内の各データフィールドを、ユニークに識別するための番号をデータフィールド番号 (DFN) と呼びます。

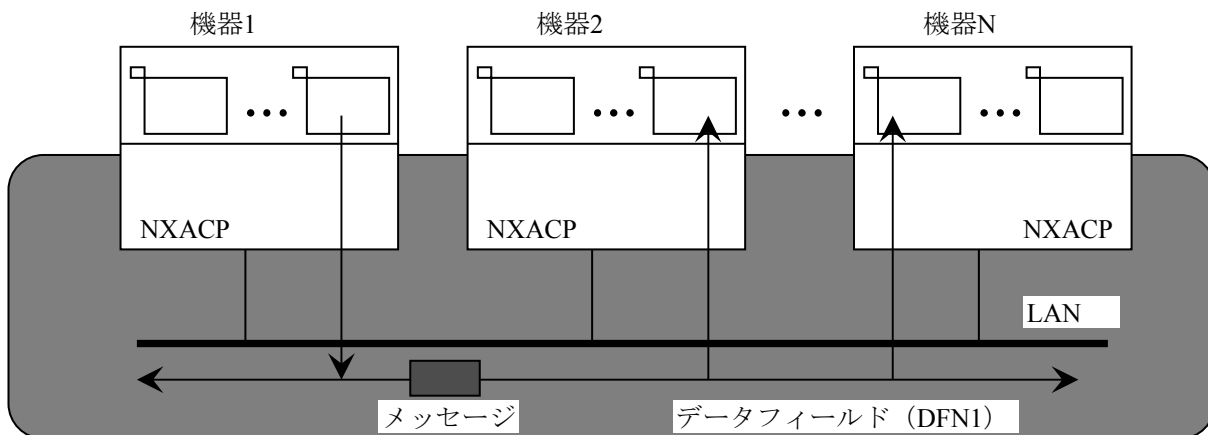


図1-5 データフィールド

データフィールドには、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があります。

- ・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です。
- ・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です。

(例) データフィールド1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で、リモートデータフィールドはデータフィールド2と3です。

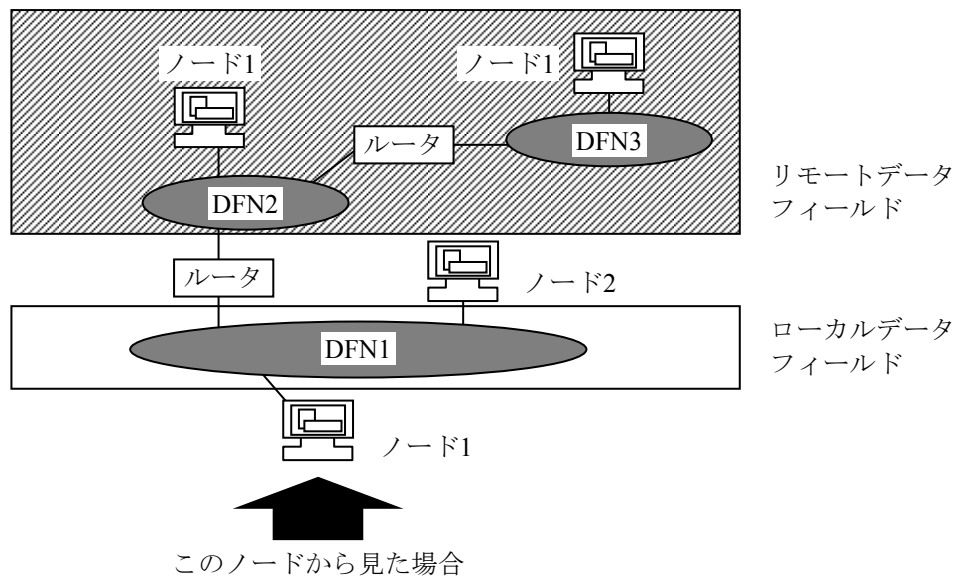


図1-6 ローカルデータフィールドとリモートデータフィールド

(2) メッセージ

NXACP間で交信するデータのかたまりの基本単位をメッセージと呼び、ユーザデータとNXヘッダで構成されます。

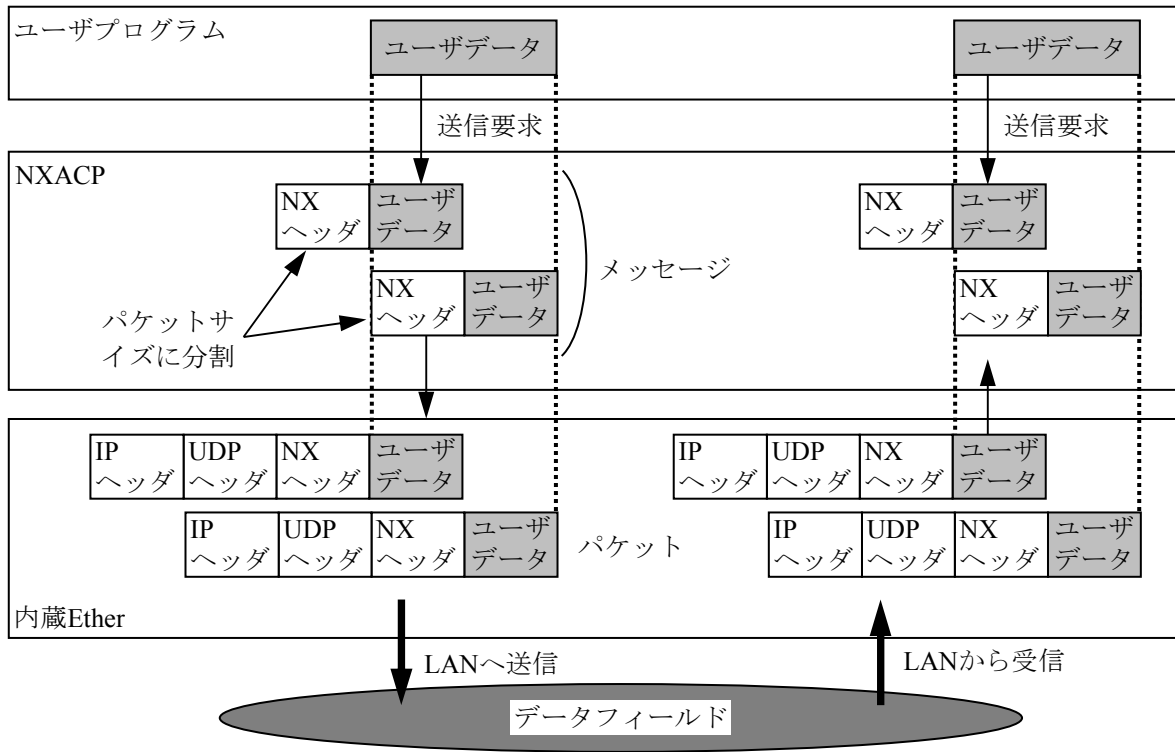


図1-7 メッセージ

(3) トランザクションコード

このシステムでは、各データフィールドを流れるメッセージをトランザクションとして処理します。そのための識別子をトランザクションコード (TCD) と呼びます。

TCDとメッセージとの対応付けは、ユーザが任意に設定でき、ユーザプログラムは、TCDを指定してメッセージを送受信します。

ユーザプログラムが指定したTCDは、各メッセージのNXヘッダに格納されます。

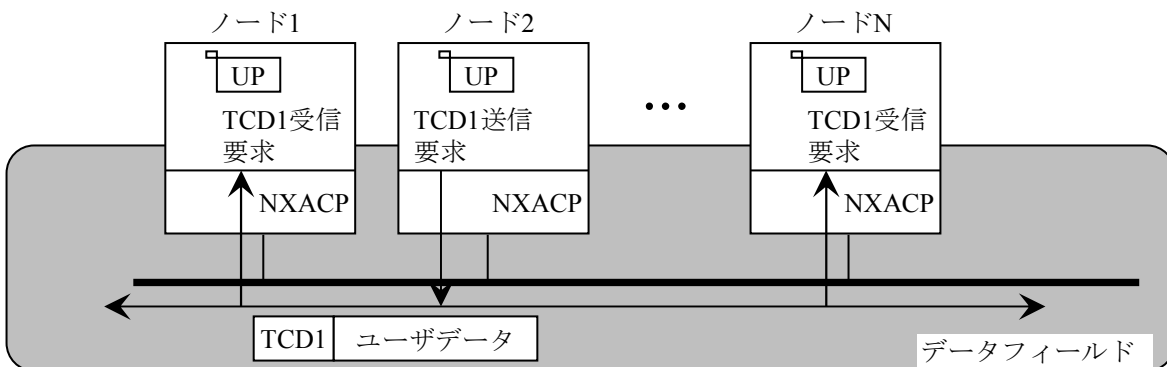


図1-8 トランザクションコード

(4) 論理ノード

データフィールドに所属する機器を論理ノードと呼びます。データフィールド内でユニーク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) です。NXACPでは、CPUを論理ノードとして扱います。

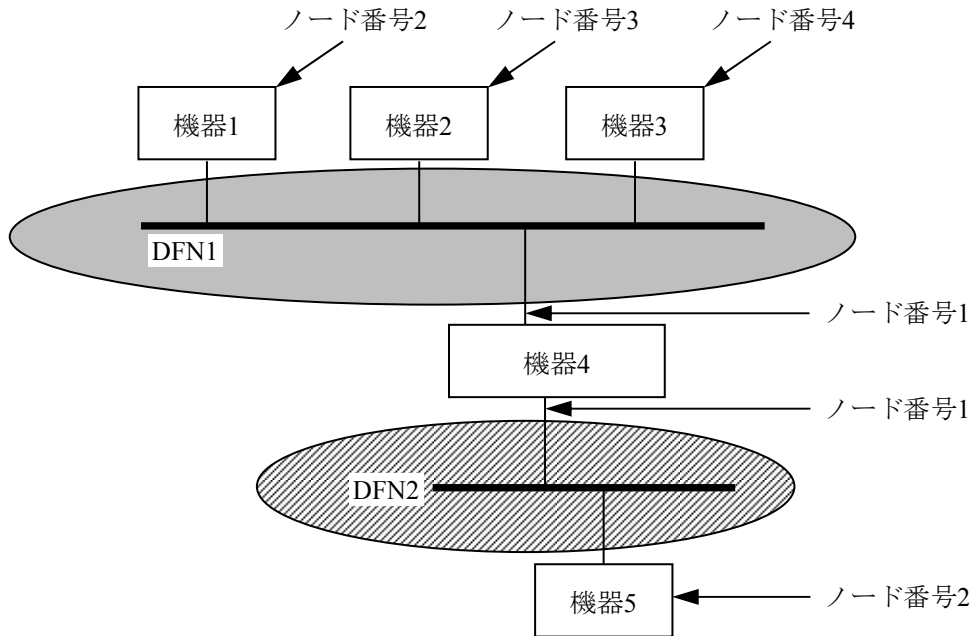


図1-9 論理ノード

ノードごとにモードを所有し、オンライン稼働中のノードをオンラインノード、テスト中のノードをテストノードと呼びます。

(5) マルチキャストグループ

マルチキャストグループとは、ある論理ノードからデータフィールド内へブロードキャストしたデータを受信するか否かをグループ化したものです。各データフィールドごとにマルチキャストグループを管理し、1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構成することができ、データフィールドに属する各論理ノードは、複数のマルチキャストグループに属することができます。

1つのマルチキャストグループは、1組のポート番号 (オンライン/テストポート番号) に対応し、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) です。

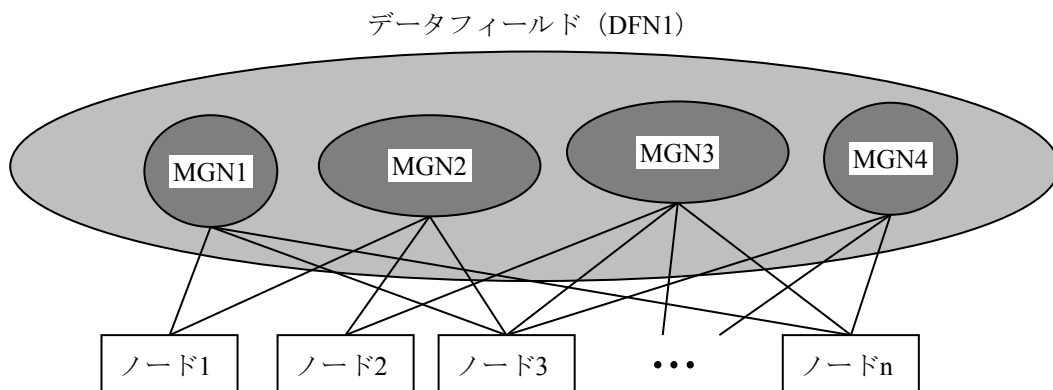


図1-10 マルチキャストノード1グループ

(6) ポート

物理的には単一の伝送路であるネットワークを使用してメッセージを送受信する場合、送受信元の属性を統一的に管理するため、送信ポートという中継点と、宛先ポートという終点を設定します。マルチキャストグループに対応するポート番号とは、宛先ポート番号です。

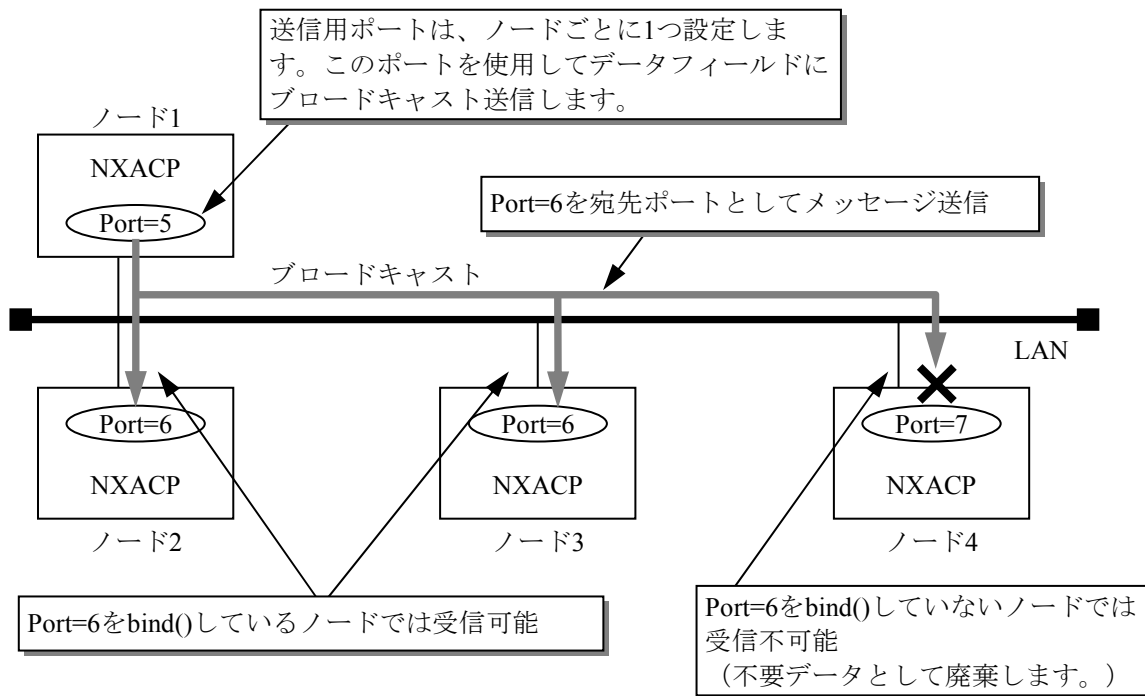


図1-11 ポート

上記ポート番号6をマルチキャストグループ1に対応させる場合、ノード2、3は、マルチキャストグループ1に属するノードということになります。マルチキャストグループ番号とポート番号の対応は、データフィールド内全ノードで統一する必要があります。

NXACPでは、送信マルチキャストグループとして宛先ポート番号を、受信マルチキャストグループとして受信ポート (bind()するポート) 番号を構築情報で指定します。

第3章 機能概要

3.1 マルチキャスト通信機能

マルチキャスト通信は、NXプロトコルのベースとなる通信方式で、以下の特長があります。

- ・複数ノードへ同時に同じメッセージを送信できるため、効率の良い送信ができる。
- ・メッセージの送受信においてノードの依存関係がなく、またコネクションの確立といった手順も不要なため、使い勝手が良い。

送信側では、送信先のデータフィールド、マルチキャストグループ、トランザクションコードなどを指定してメッセージを送信し、受信側の各ノードは自律的に必要なメッセージだけを取り込みます。

したがって、「通信相手の認識」および「通信相手との同期」などが不要となり、各機器または各ユーザプログラムの独立性を向上すると同時に、システムの拡張性を高めることができます。

マルチキャスト通信の送信範囲は、指定データフィールド番号に対応したネットワークセグメントにブロードキャストします。受信側ノードでは、マルチキャストグループを定義していない場合やトランザクションコードを定義していない場合は、不要メッセージとして廃棄します。

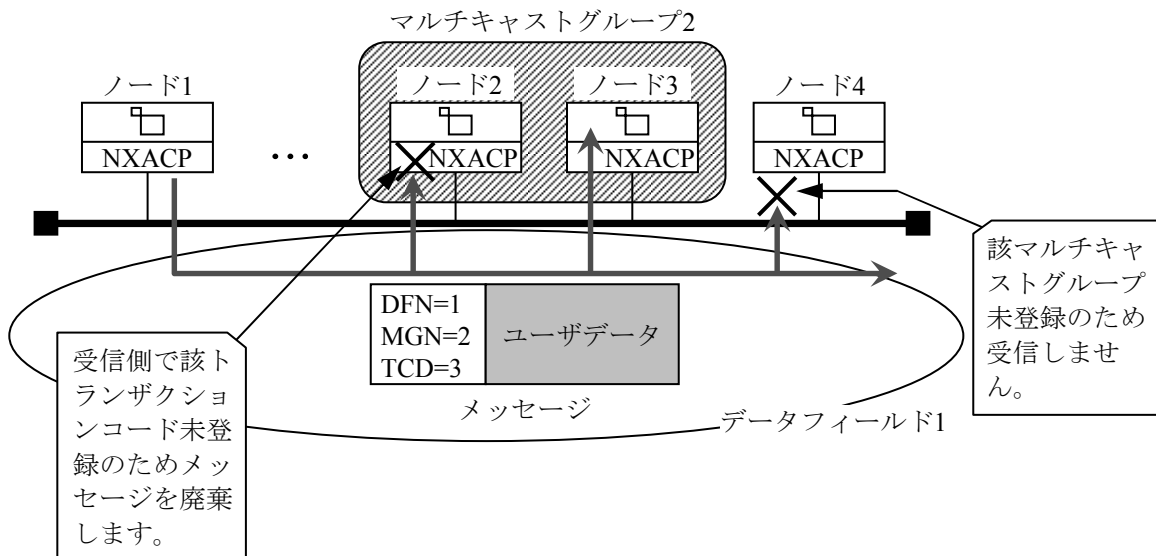


図1-12 マルチキャスト通信

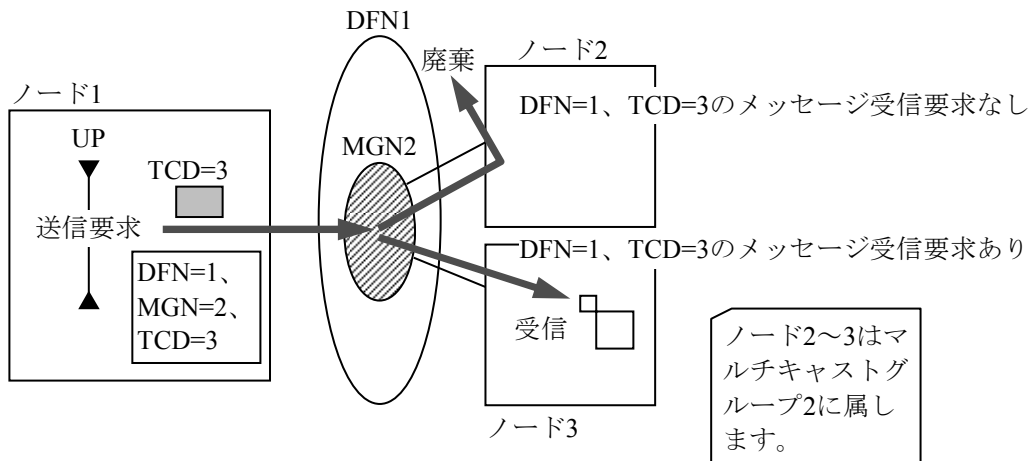


図1-13 マルチキャストグループ

ユーザは、送受信するTCDを指定してnx_put()、nx_get()マクロを発行することでメッセージを送受信できます。このとき、ユーザデータの扱いには、以下のような特長があります。

(1) 16KBデータ送受信

UPでは、最大16KBのデータを1回の送受信要求で指定できます。実際にネットワーク上を流れるパケットは、ネットワークのフレーム (MTU) に分割して送信されますが、このときの分割・再生をNXACPで制御します。

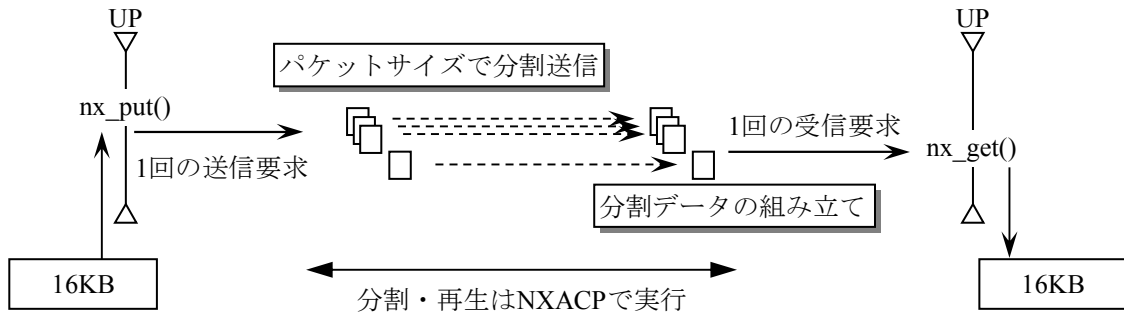


図1-14 メッセージの分割・再生

(2) 送受信範囲

メッセージの送信範囲は、データフィールド番号によって異なります。

DFN=0を指定して送信要求した場合、自ノード内専用送信になります。

DFN≠0を指定して送信要求した場合、ネットワーク専用送信になり、自ノード内に送信することはできません。

メッセージを受信する場合も同様に、自ノード内からのメッセージについては、DFN=0を指定した場合に受信できます。ネットワークで接続された他ノードからメッセージを受信する場合は、DFN≠0を指定して受信します。

なお、送信時は、nx_put()マクロのパラメータでデータフィールド番号を指定してください。

受信時は、構築情報で指定してください。

(3) TCD共用とデータ切り出し受信

1つのメッセージは、複数のUPで受信できます。したがって、伝送効率を上げるためには、送信側では、MTUサイズ未満の小さい業務データを数回出すより、複数の業務データを1つのパケットサイズ程度 (=MTUサイズ) にパックして送信し、受信側では、複数の業務データを詰めた1つのメッセージを、複数のUPで受信することを推奨します。

さらに、各受信UPは、それぞれに必要なデータだけを切り出して受信することができます。そのため、本当に必要な部分だけを切り出すことで使用メモリの縮小、データコピー時間の短縮が可能です。

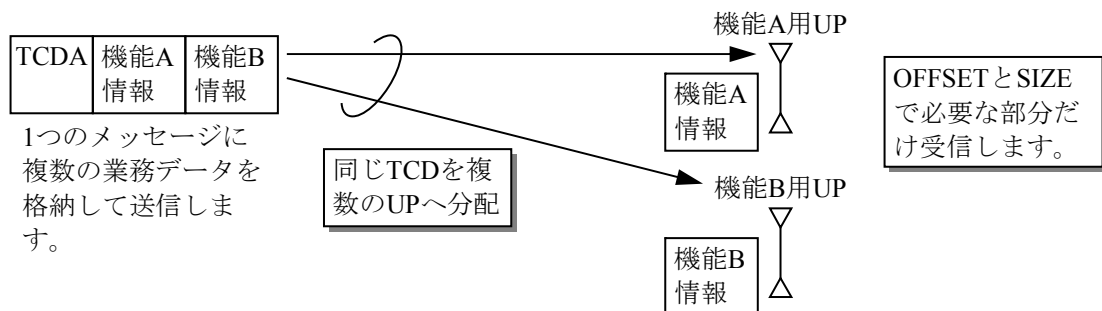


図1-15 メッセージの切り出し受信

【補足：フィルタリング機能について】

S10VEでは、RCTLNETで不要メッセージの廃棄処理をします。これを、フィルタリング機能と呼びます。

フィルタリング機能とは、必要な受信メッセージを判別し、不要なメッセージを廃棄する機能のことで、CPUの負荷低減を目的としています。

S10VEでのフィルタリング機能の対象は、ポート番号となります。したがって、NXACPを使用するユーザには、マルチキャストグループ番号がフィルタリング機能の対象となります。

NXACPでユーザが構築情報を定義していないマルチキャストグループ番号宛のメッセージを、RCTLNETが廃棄します。

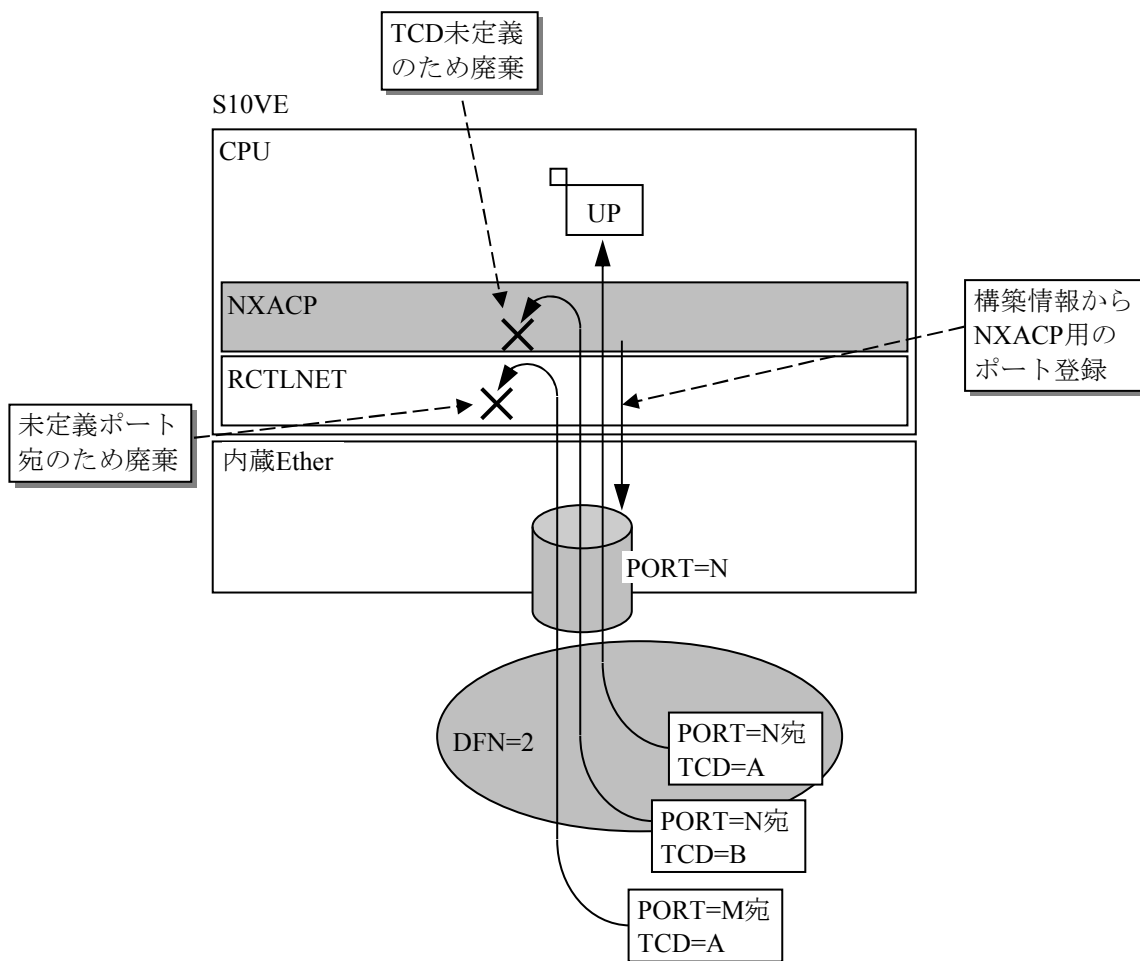


図1-16 メッセージのフィルタリング

3. 2 データフィールド管理機能

データフィールド管理とは、データフィールドに接続しているノードの接続状態を監視または自ノードの状態を他ノードへも通知する機能です。

NXACPでは、データフィールド単位にそのデータフィールドに属する各ノードが、定期的に“生存信号”というメッセージを送信します。このメッセージによって、各ノードとデータフィールドの接続状態を他ノードから監視でき、システム全体の接続状態が管理できるようになります。

● 生存信号の送出機能

NXACPでは、ユーザからの立ち上げ要求を受け付けた時点から、生存信号メッセージの定周期送信をします。生存信号には、ノードモードやシステム立ち上げ時間、IPアドレス、生存信号モードなどの情報を付加して送信します。生存信号メッセージの送信周期は、構築情報でユーザが指定します。

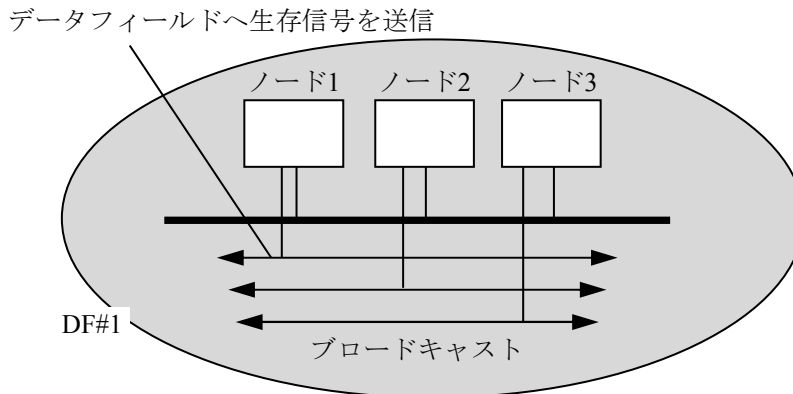
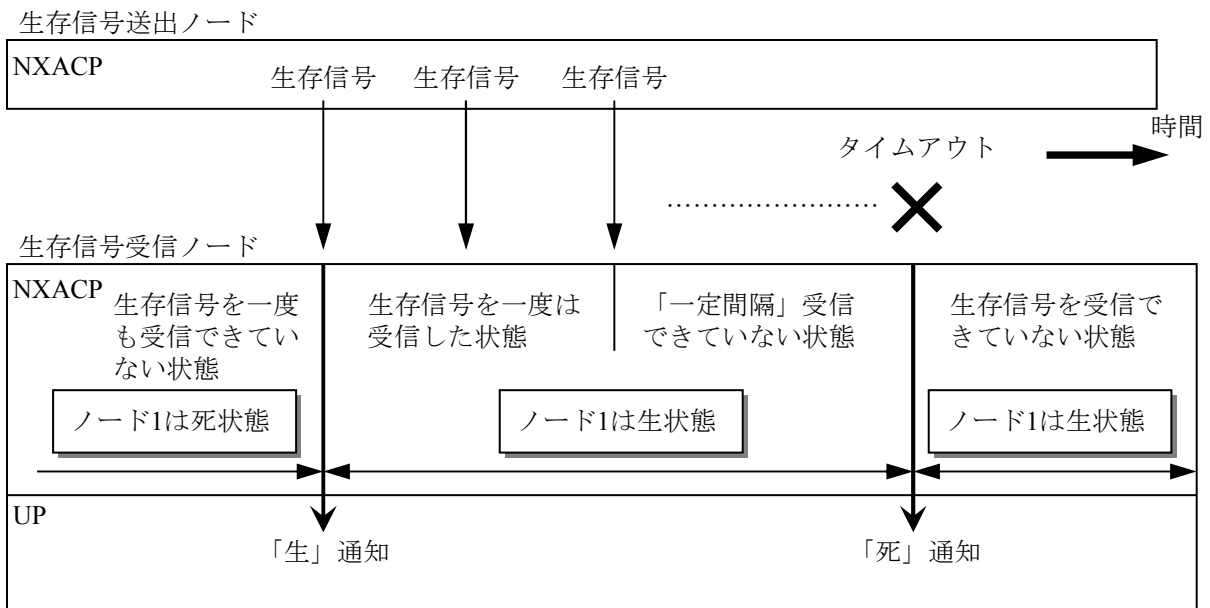


図1-17 生存信号

● ノード接続状態の監視機能

データフィールドに接続する他ノードの接続状態を監視し、その状態に変化が発生するとユーザへ通知する機能です。通知する状態変化には、“生：生存信号受信開始”、“死：生存信号中断後タイムアウト監視時間超過”があります。また、`nx_dfsts()`マクロでカレントの状態を参照することもできます。



生存信号は、二重化LAN構成においては、LANごとに送信／監視します。

図1-18 ノード状態管理

NXACPでは、生存信号はユーザからの立ち上げ要求以降自動で送出しますが、他ノードの状態の監視は、ユーザの指定（構築情報）となります。これは、コントローラでの生存信号受信およびノードの状態変化監視の負荷を低減できるようにするためです。他ノードの状態変化を監視する必要のないノードではノード接続状態の監視機能をOFFで指定し、動作させることを推奨します。

WSやPCが接続するネットワークシステムでは、ノードの状態変化をできるだけWSまたはPCで監視し、個々のコントローラでは、生存信号は受信しない運用を推奨します。

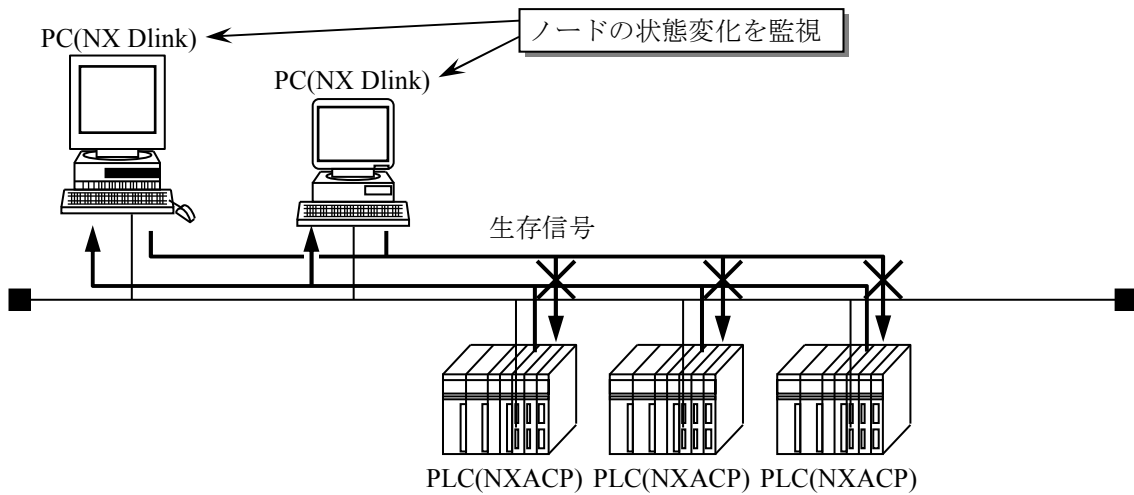


図1-19 状態監視ノード (1)

WSやPCが接続しないネットワークシステムでノードの状態変化を監視する必要がある場合、特定コントローラでノードの状態変化を監視し、他のコントローラでは、生存信号は受信しない運用を推奨します。また、この場合、ノードの状態変化を監視するコントローラでは、データフィールド監視業務に専用化して使用することを推奨します。

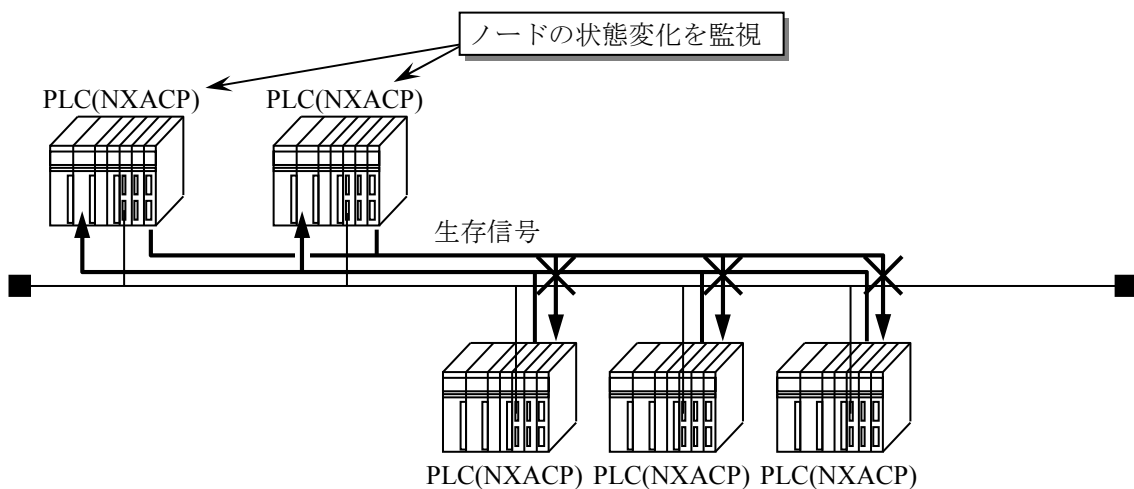


図1-20 状態監視ノード (2)

3.3 二重化LAN制御機能

NXACPでは、データフィールドを構成するネットワークを二重化できます。データフィールドの構成LANを二重化することでデータ通信の信頼性を向上することができます。この機能を使用するには、構築時のデータフィールド定義で二重化LAN関連の情報を設定する必要があります。

データフィールドの構成LANを二重化した場合でも、ユーザからは、下図に示すように物理的に2本のLANが1つのデータフィールドに見え、ユーザは2本のLANを意識する必要がありません。

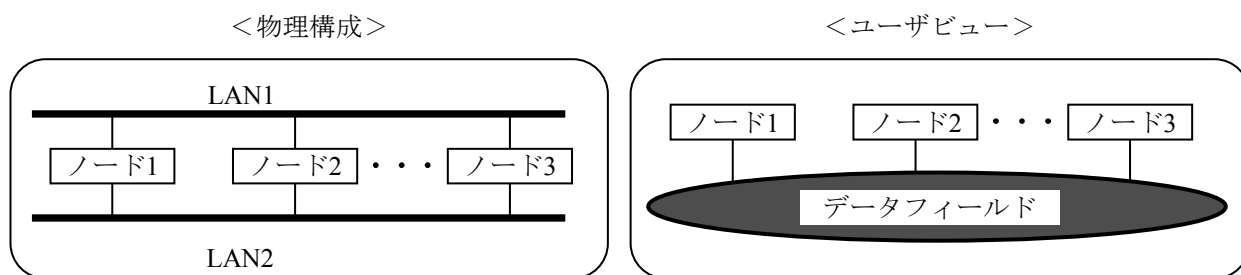


図1-21 LANの二重化

NXのマルチキャスト通信は、二重化LANを使用した場合、デュアル形態となり、ユーザはネットワークのマスタ/スレーブなどの複雑な構成管理から解放されます。

この方式には、以下のような特長があります。

- ・両方のLANへメッセージを送信します。
- ・両方のLANからメッセージを受信し、重複を検知した場合、不要メッセージを廃棄します。
(先着受信、後着廃棄)
- ・ネットワークのマスタ/スレーブをユーザが管理する必要はありません。

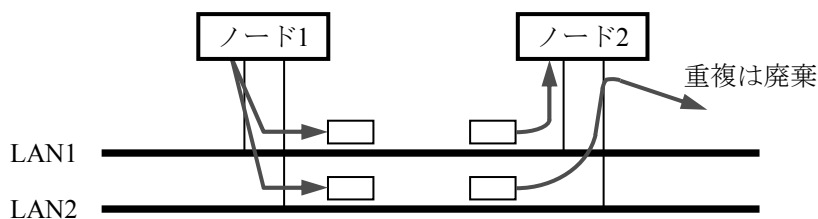


図1-22 二重化LAN通信方式

NXACPが周期送信する生存信号も、上記マルチキャスト通信と同じデュアル形態となり、各ノードのデータフィールド接続監視はLANごとに監視されます。

【留意事項】

二重化したLANにおいて、どちらか一方のLANを指定して送受信することはできません。

3.4 テスト機能

テスト機能とは、オンライン稼働中のシステム上で、論理ノードごとのテストを実現する機能です。

オンライン稼働中のノードをオンラインモードノード（以降、オンラインノードと略します）、テスト中のノードをテストモードノード（以降、テストノードと略します）と呼び、テスト対象ノードをテストモードに設定することで、オンライン系のノードに擾乱を与えないテストが可能となります。

これは、データフィールドを流れるメッセージにも、オンラインノードが送信するメッセージをオンラインモードメッセージ（以降、オンラインメッセージと略します）、テストノードが送信するメッセージをテストモードメッセージ（以降、テストメッセージと略します）というように、メッセージにもモードを付けることによって、以下のようなテスト形態を実現しています。

- ・テストノードから送信するメッセージは、テストメッセージとして送信され、オンラインノードでは、オンラインメッセージだけを受信します。したがって、オンライン環境でオンライン系のノードに擾乱を生じさせないテストが実施できます。
- ・オンラインメッセージをテストノードでは受信可能と設定することで、オンラインデータを使用したテストが実施できるようになります。

NXACPでは、テストノードで受信するメッセージモードを下記2つの中から選択でき、上記以外にも多様なテストシステム構成を構築できます。

- ・オンラインメッセージだけを受信するモード
- ・テストメッセージだけを受信するモード

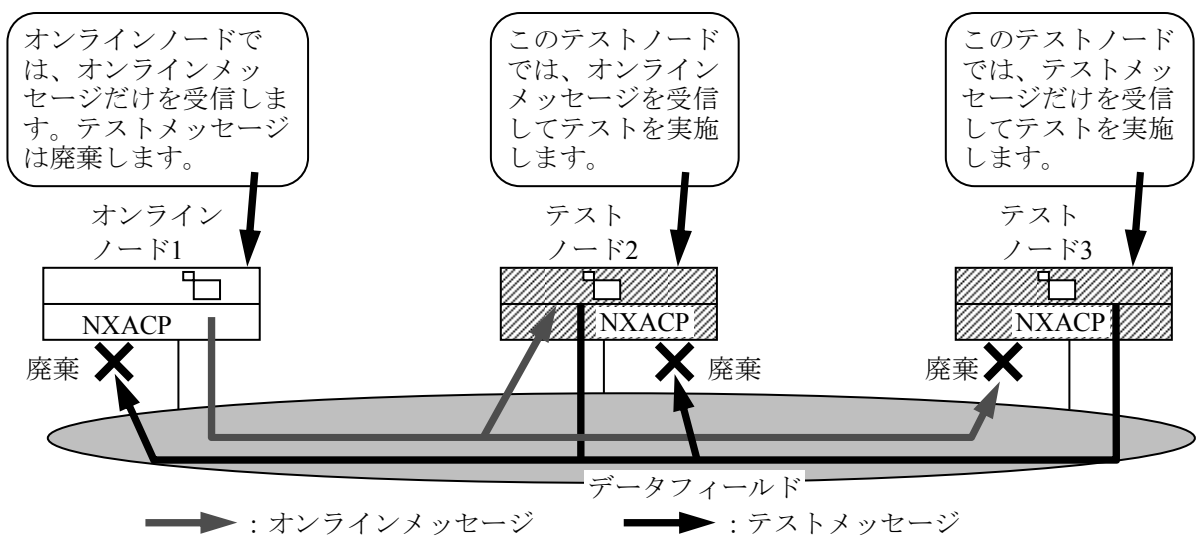


図1-23 テスト機能

【補足】

オンライン系のノードへは、上記のとおりテストメッセージの入力を抑止するため、擾乱を与えませんが、ネットワークへは、テストメッセージが送出されますので、ネットワーク負荷を考慮して使用してください。

また、ネットワーク共有メモリ機能使用時は、ノードモードとメッセージモードによるメッセージ入出力制御の対象とはなりませんので、注意してください。ソフト転写については、ノードモード、メッセージモードによるメッセージ入出力制御の対象となります。

3.5 システム管理機能

このシステムには、以下のシステム管理機能があります。

- ・ 障害通知／ログ機能
- ・ DHP取得／制御トレース機能
- ・ ユーザタスク状態管理機能

● 障害通知／ログ機能

次に示す各種障害は、エラーログ、EAS、または障害通知専用IRSUBでユーザに報告します。ユーザでは、障害発生をタイムリーにキャッチし、的確な対処を効じることが可能です。

・ 通信障害

NXプロトコル障害やソケットの障害について通知します。ハードウェア障害については、RCTLNETが通知します。

・ バッファなどリソースの使用率変化

ユーザが構築情報で指定するNXACPのリソース（バッファなど）使用状態が、しきい値を超えた場合やオーバフローした場合、その状況を通知します。

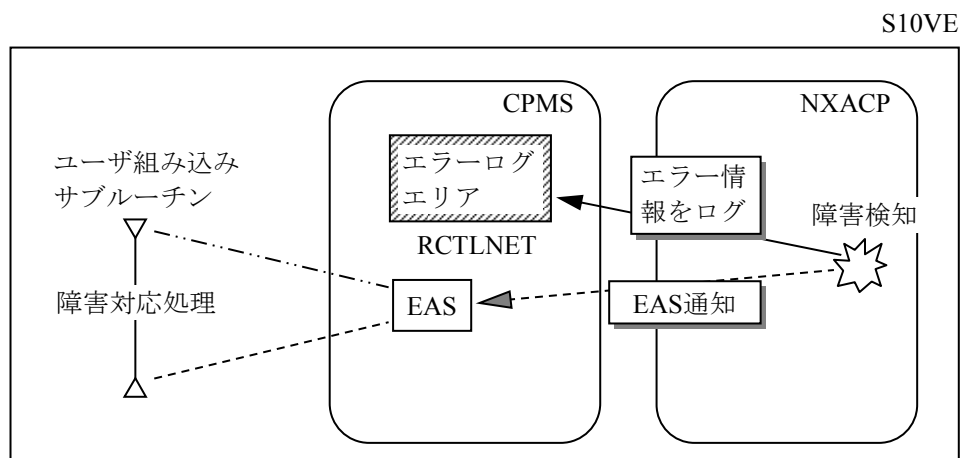


図1-24 障害通知

なお、ノード接続状態変化については、変化通知専用IRSUBにリンクします。他ノードの生存監視機能を使用した場合、LANごと、ノードごとにDFへの接続状態変化を通知します。

● DHP取得／制御トレース機能

ある決められた処理ポイントにおいて、そのポイントを通過したことを、CPMSが管理します。

DHPバッファに逐次記録します。DHPバッファに記録された情報は、エラー発生時エラー情報と合わせて参照したり、コマンドで参照できますので、障害発生時、他サブシステムへのアクセスと合わせて、ユーザタスク動作を解析できます。

また、送受信したメッセージの制御情報をトレースバッファに逐次記録します。トレース機能の使用有無およびバッファエリアサイズは、構築時にユーザが指定してください。

S10VE

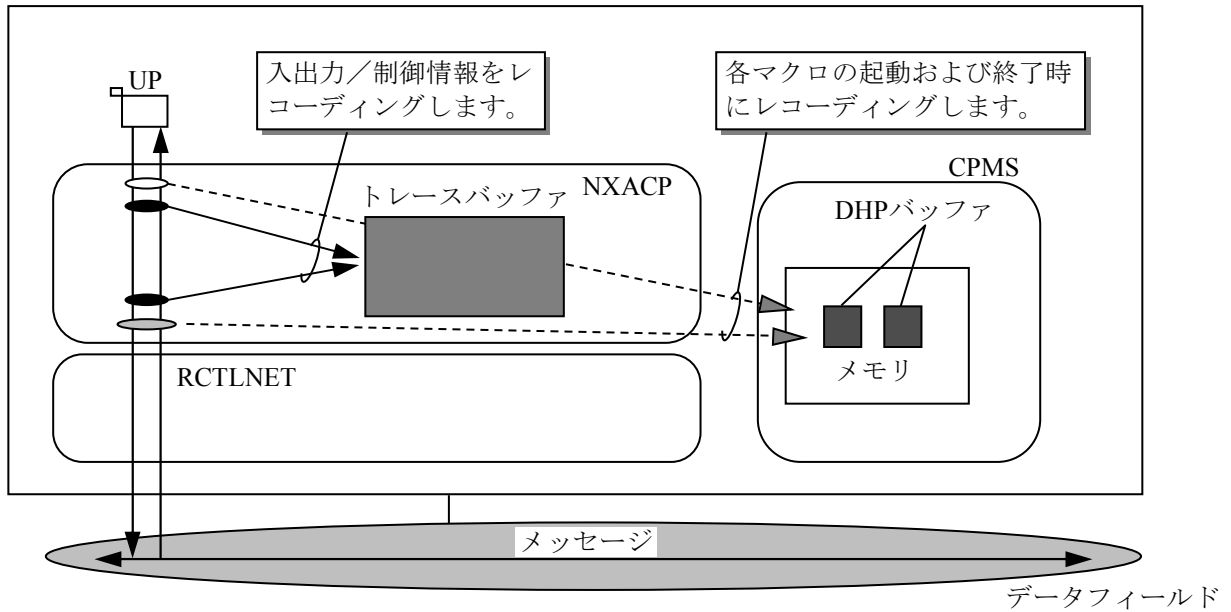


図1-25 トレース

● ユーザタスク状態管理機能

メッセージを受信するユーザタスクの状態を監視します。これは、メッセージを受信できない場合に、バッファオーバーフローを防止するためのものです。

NXACPでは、ユーザタスクの状態をメッセージ到着時にチェックし、DORMANT/IDLE/NON-EXISTの場合、該当タスクへはメッセージの配信をしません。また、ユーザタスクがDORMANT遷移時でも、該当タスクが受信処理中および受信予定のメッセージを廃棄します。

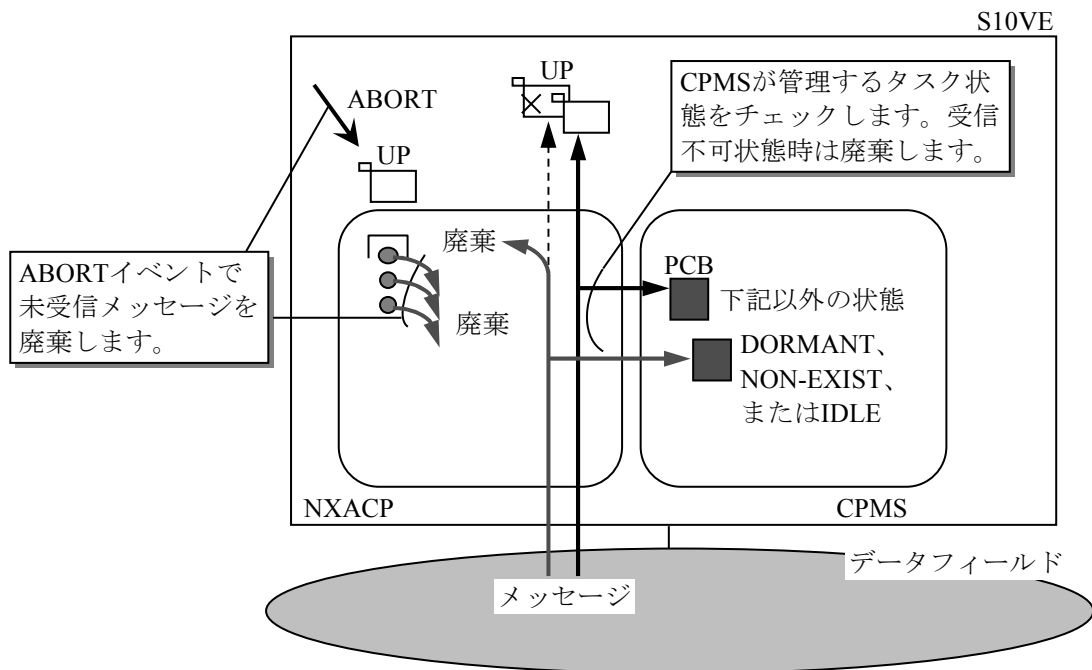


図1-26 UPタスク管理

3. 6 運用管理機能

NXACPのサービス開始/停止は、ユーザからの要求で制御できます。

NXACPは、`nx_init()/nx_quit()`を発行することで起動/停止をします。

NXACPの起動/停止とは別に、データフィールドのサービス開始/停止は、`nx_dfup()/nx_dfdwn()`を発行することで行えます。

(1) NXACP起動

ユーザは、システム立ち上げ時に`nx_init()`を発行してください。`nx_init()`の発行によって、NXACPはデータフィールドごとのサービス開始要求を受け付ける準備をします。

(2) データフィールドのサービス開始

データフィールドのメッセージ送受信開始要求は、データフィールドごとに、`nx_dfup()`を発行してください。また、`nx_dfup()`は、ネットワークに対応しない自ノード内専用データフィールドに対しても発行する必要があります。

したがって、複数のデータフィールドを定義した場合は、`nx_dfup()`を複数回データフィールドごとに発行する必要があります。なお、データフィールドを二重化LANで構成している場合でも、`nx_dfup()`は1度だけ発行してください。

(3) データフィールドのサービス停止

システムを動作したまま特定のデータフィールドの送受信だけを停止したい場合は、`nx_dfdwn()`をデータフィールドごとに発行してください。この機能を使用することでシステムを停止することなく、ノードモードを変更できます。自ノード内専用データフィールドに対しても、送受信を停止したい場合は、`nx_dfdwn()`を発行してください。

`nx_dfdwn()`は、要求されたデータフィールドがローカルデータフィールドの場合、他ノードに計画停止 (SHUTDOWNモードの生存信号) であることを通知してから送受信サービスを停止します。`nx_dfdwn()`でデータフィールドのサービスを停止させた後再起動する場合は、再度`nx_dfup()`を発行してください。

(4) NXACP停止

システムを動作したままNXACPまたは全データフィールドを一度に停止したい場合は、`nx_quit()`を発行してください。`nx_quit()`を発行した場合も、`nx_dfdwn()`同様に他ノードへ計画停止通知であることを通知してから送受信サービスを停止します。

`nx_quit()`でNXACPを停止させた後再起動する場合は、`nx_init()`を発行してください。

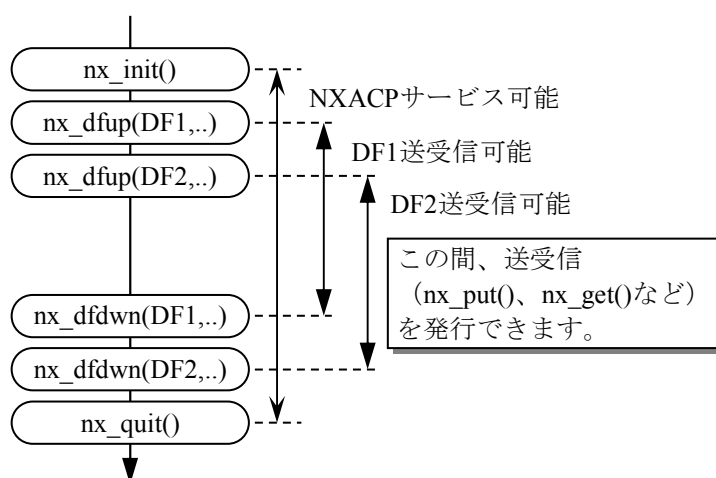


図 1-27 NXACP運用管理

3.7 ネットワーク共有メモリ機能

ネットワーク共有メモリとは、メモリエージでデータフィールドの属する各ノードをネットワーク結合するもので、定周期で各ノードごとの書き込みデータを読み出し側に転写する機能です。

同一メモリアドレスに対しては、ノード1台が書き込み可能で、複数台のノードで読み出せるようになります。

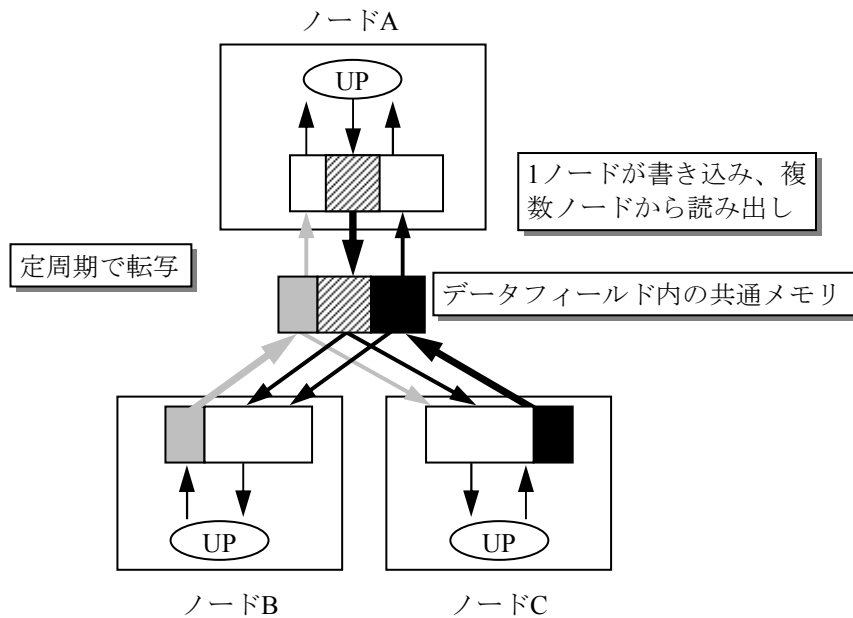


図1-28 ネットワーク共有メモリ

ソフト転写でミリ秒の転写周期をサポートします。

NXACPでサポートするネットワーク共有メモリには、以下の特長があります。

- ・通信手続きの隠蔽

ユーザはメモリインタフェースで通信ができます。

- ・ソフト転写による低価格化

イーサネットでのソフトウェアによる転写メモリ機能をサポートすることで、特殊なハードウェアを購入しなくても容易にメモリインタフェースの通信ができます。

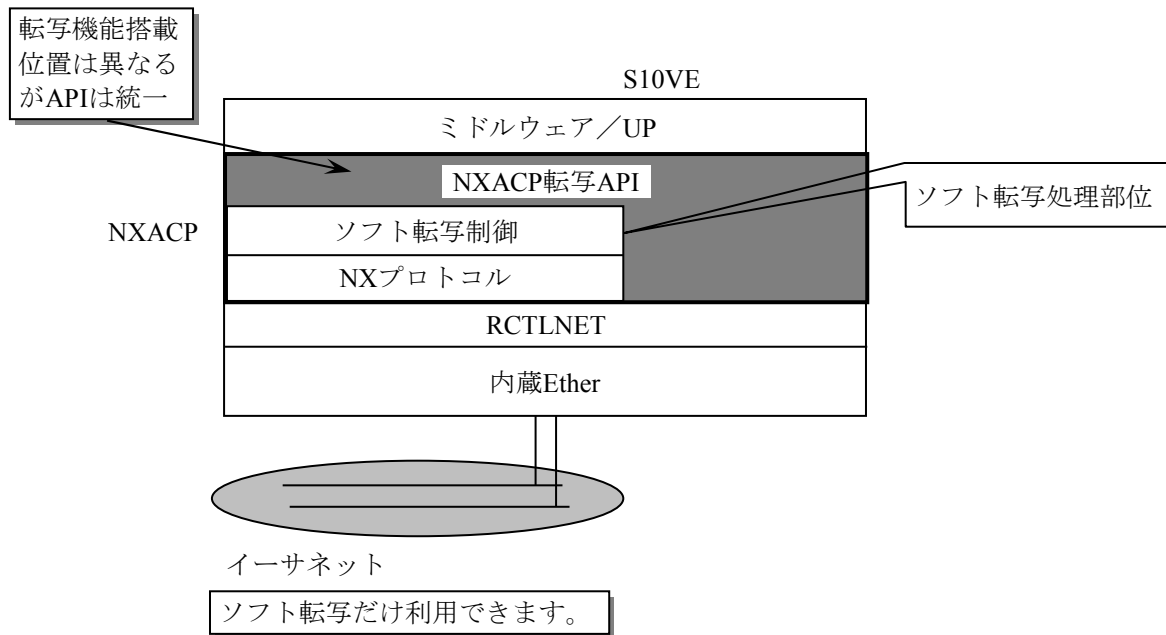


図1-29 共有メモリの特長

【補足】

NXACPのネットワーク共有メモリ機能は、他機能同様ラダーインタフェースはありません。

3.8 システム構築機能

NXACPの構築は、POC上に構築情報を定義し、ターゲットとなるRPDP管理サイト情報にNXACPの構築コマンドを使用してローディングします。

S10VEへのリモートローディングは、RPDPのsvrpl（リモートローディング）コマンドを起動してください。

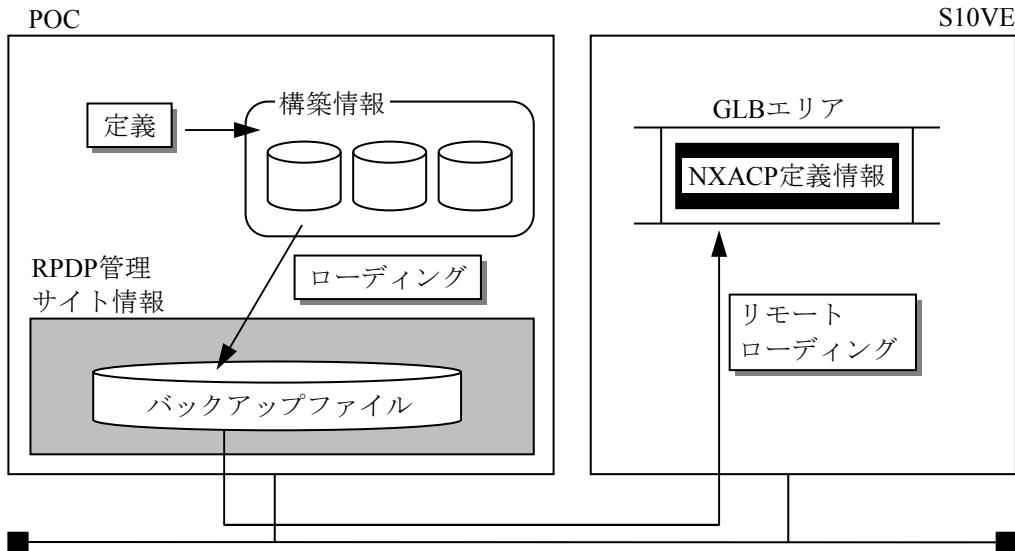


図1-30 システム構築

S10VEでは、マルチプロセッサ構成（機能分散型）タイプのサイトをRPDPのジェネレーションで構成することができます。

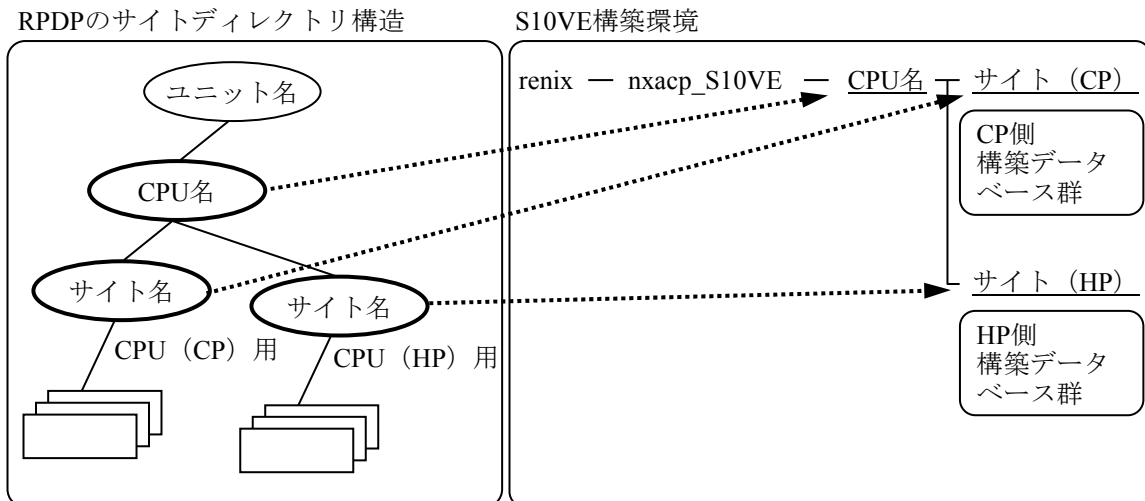


図1-31 NXACPの構築環境

構築データベース群には下記の構築情報があります。

- バッファ面数やサイズなどのバッファ情報
- 生存信号送出間隔、生存信号受信有無などの生存信号情報
- ネットワークアドレス、ポート番号などのネットワーク情報
- TCD、MGN、送受信タスク番号などのトランザクション情報

これらの情報を、データベースファイルに定義し、NXACP構築情報ローディングコマンドを使用してコンパイルし、ターゲットとなるサイトにローディングします。NXACP構築情報ローディングコマンドは、RPDPコマンドを使用してGLBエリアにローディングしますので、システムジェネレーションが完了した後に行ってください。

なお、オンライン変更はサポートしていません。構築情報をターゲットマシンに反映する場合、構築情報ローディングコマンドは、該当マシンのNXACPを一時停止させた状態にします。

この場合、SHUTDOWN予告の生存信号は送信しませんので注意してください。

このページは白紙です。

第2編 機能ガイド

第1章 マルチキャスト通信機能

1. 1 通信の特長

このシステムで利用できる通信方式について説明します。

1. 1. 1 メッセージ通信の単位

このマニュアルでは、ユーザが入出力する情報の固まりを「データ」と呼びます。

ユーザが指定したデータにNXヘッダを付加し、NXACP内で1つの論理的な情報として扱うものを「メッセージ」と呼びます。また、ネットワーク上を流れる1つの情報単位を「パケット」と呼びます。

NXACPは、ユーザから指定されたデータにNXヘッダを付加（メッセージ化）し、それをパケットに分割してネットワーク送信します。受信側のNXACPでは、分割されたパケットをメッセージに組み立て、NXヘッダを削除してユーザに渡します。

ユーザが指定できる最大データ長は16KBです。

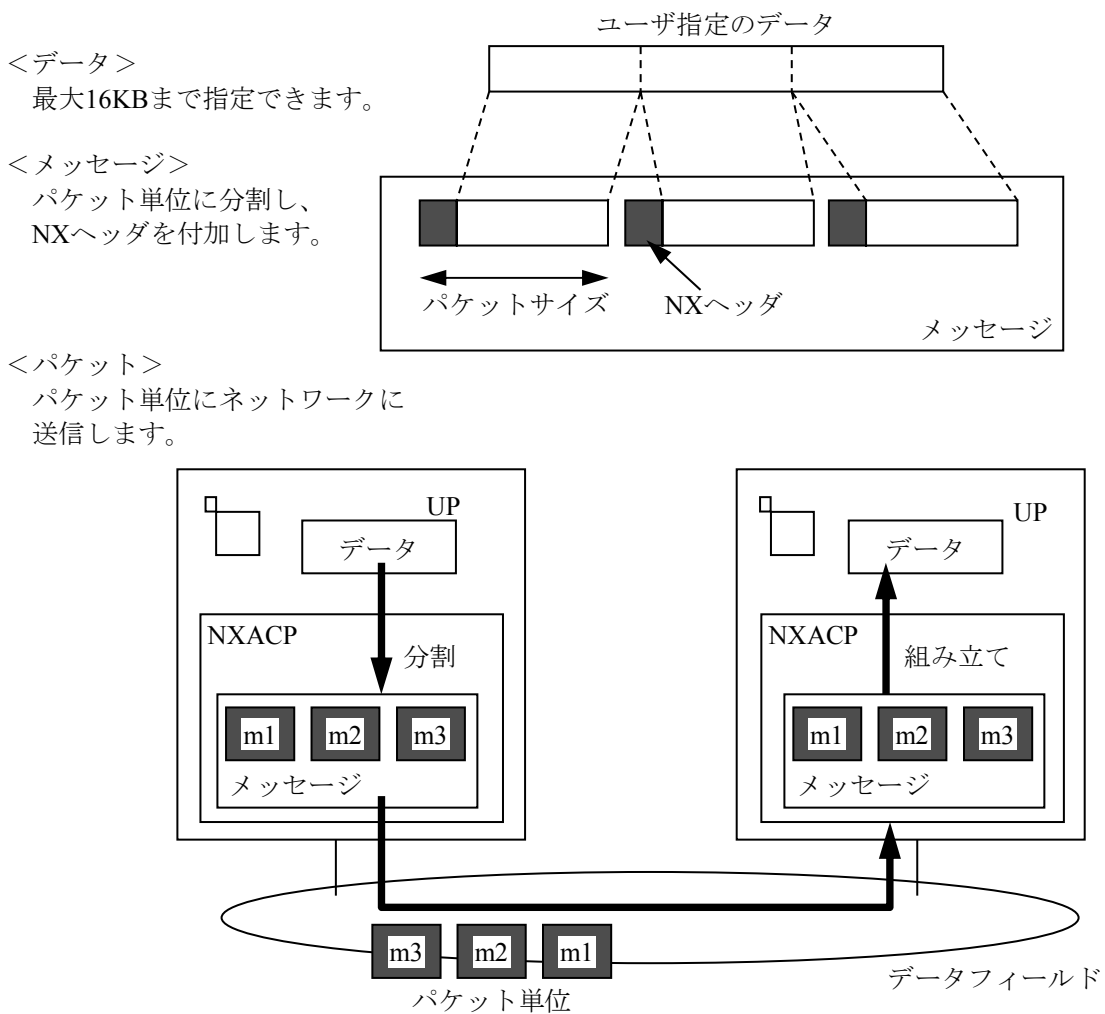


図2-1 通信単位

1. 1. 2 メッセージの分割・組み立て

ユーザが送信時に指定できる最大データ長は16KBですが、NXACPでは、パケットサイズに分割して送信し、受信側で組み立てます。

分割するパケットサイズは、ユーザが構築情報でデータフィールドごとに指定するバッファ1ケースサイズにNXヘッダ（64バイト）を加えたサイズとなります。

指定可能な最大バッファ1ケースサイズは、以下のとおりネットワーク種別にかかわらず共通です。

表2-1 バッファサイズ

ネットワーク種別	最大バッファ1ケースサイズ	パケットサイズ
イーサネット（内蔵Ether）	1408バイト	1472

NXヘッダサイズは64バイト固定です。したがって、1パケット内ユーザデータ長は、最大パケットサイズから64バイトを引いた値になります。

以下に、送信時にユーザが指定したデータサイズとパケット使用ケース数の関係を示します。

$$\text{パケット使用ケース数} = \frac{(\text{ユーザ指定データサイズ}-1)}{\text{バッファ1ケースサイズ}} + 1$$

つまり、最大バッファ1ケースサイズで構築し、16KBの送信要求を発行した場合、以下のようになります。

$$\text{パケット使用ケース数} = \frac{(16384-1)}{1408} + 1 = 11+1 = 12 \text{ ケース}$$

【補足】

バッファ1ケースサイズはデータフィールドごとに全ノードで統一してください。

NXACPでは、バッファ1ケースサイズより大きいパケットを受信した場合、障害情報をユーザに通知して、受信パケットを廃棄します。

1. 1. 3 ユーザプログラム構造

ユーザは、データの送受信要求時、`nx_put()`/`nx_get()`マクロを発行してください。

`nx_put()`/`nx_get()`マクロは、以下に示す特長があります。なお、このマクロを発行できるのは、タスク番号が1~208までのタスクです。

(1) 送信要求

ユーザは、データ送信要求時、`nx_put()`マクロを発行してください。

NXACPでは、ユーザプログラムの送信処理がハードウェアの処理と非同期に実行できるよう、ユーザからの送信要求を受け付けるサービスと、ネットワーク上にパケットを送信するサービスを分けています。また、ネットワーク上にパケットを送信するサービスは、周期起動（周期時間は構築情報）を採用し、スイッチング回数の低減を図っています。

したがって、`nx_put()`が正常に終了しても、ネットワークに正常に送信されたことを保証するものではありません。ユーザプログラムからの送信要求（`nx_put()`）に対し、`nx_put()`マクロが発行元ユーザに返す情報は以下の情報に限定されます。なお、以下の障害検知時は、いずれも送信処理を中断します。

- ・動作環境チェックで異常を検知したときのエラーコード
- ・パラメータ異常を検知したときのエラーコード
- ・NXACPの送信バッファが満杯状態のときのエラーコード
- ・ネットワークの障害を検知した後のエラーコード

ネットワーク上にパケットを送信するサービスを実行時にエラーを検出した場合のユーザ通知は、NXACPではなく、ネットワークドライバがEASに通知します。EASに報告する障害詳細に関しては「ユーザーズマニュアル S10VE総合編（マニュアル番号 SEJ-1-001）」を参照してください。

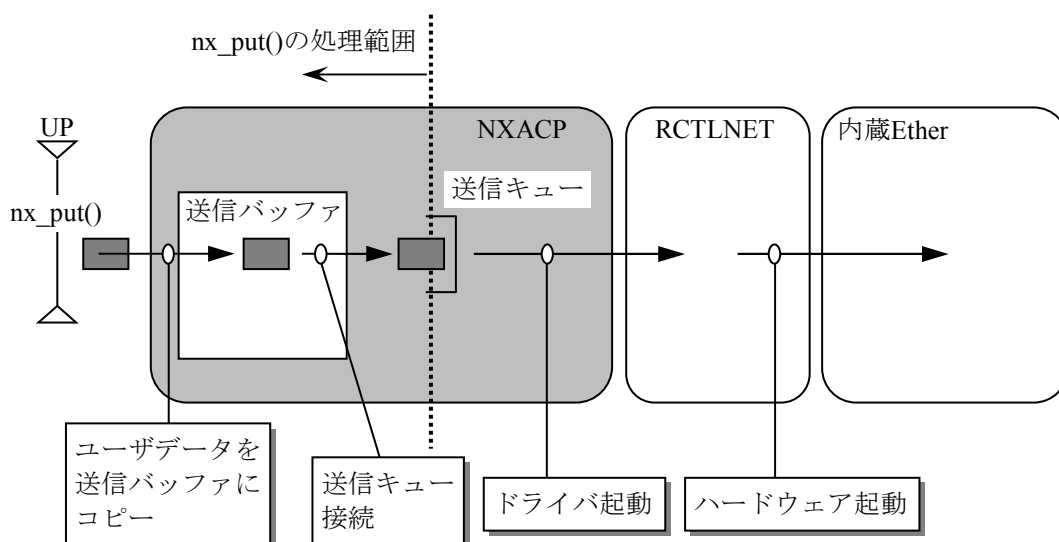


図2-2 nx_putマクロ

ユーザは、障害検知部位によっては、`nx_put()`のリターンコードでなく、EASで障害を検知する必要があります。このとき、NXACPの検知する障害だけでなく、RCTLNETの検知する障害報告も監視する必要があります。

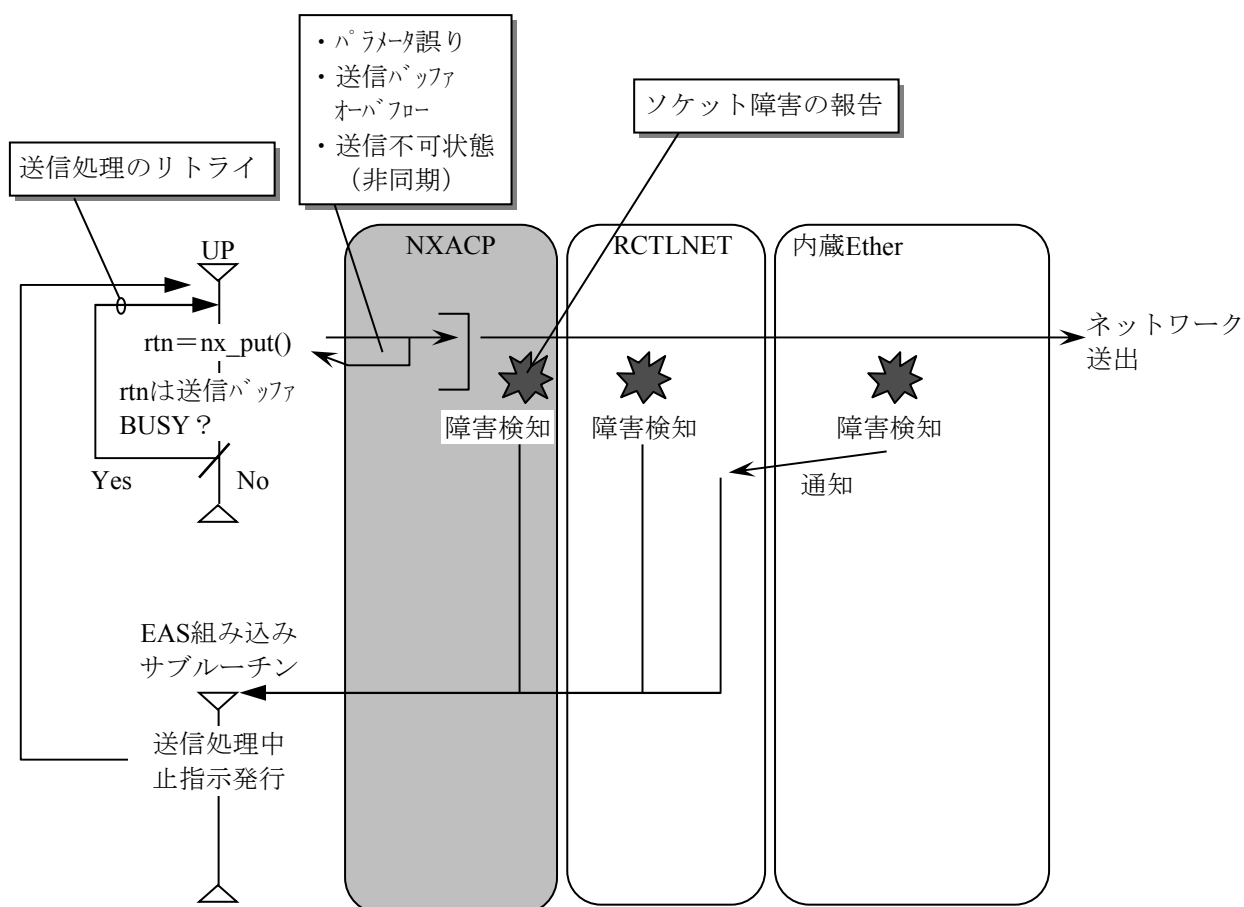


図2-3 障害情報の検知（送信）

【補足】

・ソケット障害について

内蔵Etherなどハードウェアの障害については、RCTLNETからEASにその障害内容を通知します。

NXACPがEASに通知する障害は、ネットワーク構築誤りが原因のもので、この障害をソケット障害と呼びます。NXACPがEAS通知する障害に関する詳細は「5. 1 障害通知機能」を参照してください。

・送信不可状態について

内蔵Etherなどハードウェアの障害発生に伴いネットワーク送信処理が継続不可状態に陥った場合、障害発生以降の`nx_put()`要求は、送信不可状態としてエラーリターンします。また、障害発生前に`nx_put()`で送信要求され、送信キューでネットワーク送信サービス待ちになっていたメッセージは、すべて廃棄されます。

(2) 受信要求

ユーザは、データ受信要求時、`nx_get()`マクロを発行してください。

受信処理も送信処理同様、ハードウェアからの割り込み処理とユーザプログラムの受信処理が非同期に実行できるよう、ユーザからの受信要求を受け付けるサービスと、ネットワークからの割り込みサービスを分けて処理します。

したがって、受信要求時`nx_get()`発行元ユーザに障害情報として通知できるのは、以下の場合だけで、いずれも受信処理を中断します。

- ・動作環境チェックで異常を検知したときのエラーコード
- ・パラメータ異常を検知したときのエラーコード
- ・規定時間内にメッセージ到着なしのエラーコード（規定時間はUPのパラメータ指定）
- ・NXACPの停止を検知したときのエラーコード

メッセージ受信処理中のRCTLNETや内蔵Etherのハードウェアでの障害については、RCTLNETでEASに通知します。なお、RCTLNETや内蔵Etherのハードウェアでの障害時でも、`nx_get()`はエラーリターンしませんので注意してください。

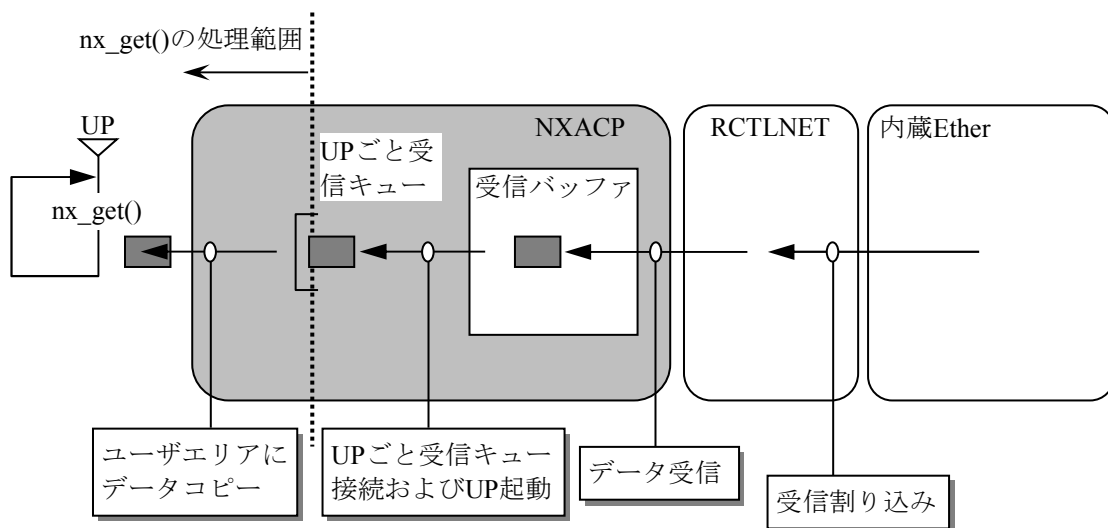


図2-4 `nx_get`マクロ

`nx_get()`を発行するユーザタスクはループ型の構造を取り、ループの先頭でトランザクションの受信要求を行うことを推奨します。これは、`exit()`や初期起動より、`nx_get()`で受信メッセージの到着を待ち、メッセージ到着時NXACPから起動してもらう構造の方がオーバーヘッドが少ないからです。

ユーザは、障害検知部位によってはnx_get()のリターンコードでなく、障害をEASでタイムリーに検知する必要があります。このとき、NXACPの検知する障害だけでなく、RCTLNETの検知する障害報告も監視する必要があります。

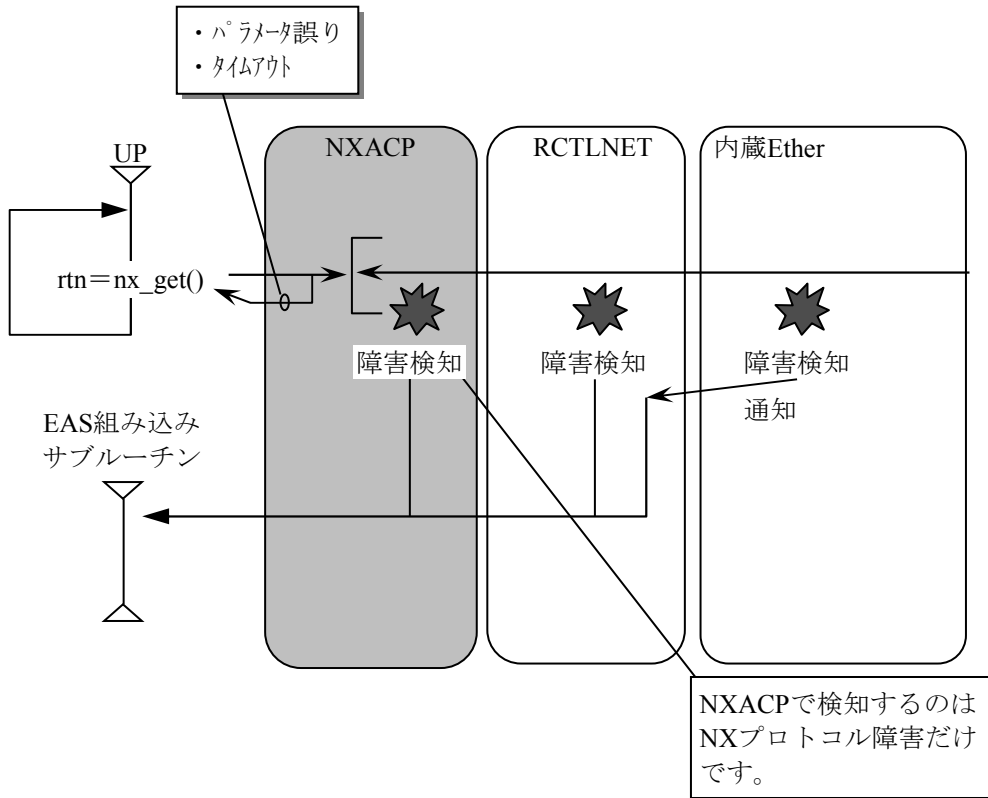


図2-5 障害情報の検知（受信）

1. 1. 4 メッセージ処理順序

NXACPでのメッセージ送受信処理では、送受信ともメッセージ発生順番（FCFS）でサービスします。したがって、メッセージ送信においては、同じタスクレベルでは、ユーザの送信要求順にサービスとなり、メッセージ受信においては、マルチキャストグループ単位での到着順サービスとなります。

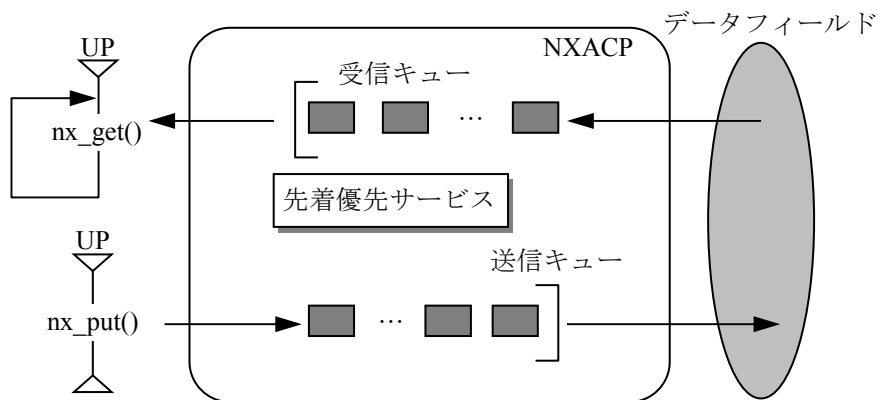


図2-6 処理順序

【補足】

NXACPは、メッセージ優先制御はサポートしていません。

NXプロトコルには、「メッセージ優先レベル」といって、メッセージ送受信処理にプライオリティを付け、プライオリティの高いメッセージを優先してサービスするという機能があります。

この機能は、例えば、オンラインメッセージには高いプライオリティを割り付け、ファイル転送データには低いプライオリティを割り付けることでファイル転送をオンラインメッセージ処理の空いた時間に実行できるという利点があります。しかし、コントローラでは、ファイル転送タイプのデータはないことから、この機能はサポートしていません。

NXACPは、ネットワークにメッセージ送信時、ヘッダ内送信レベルに0（ゼロ）を設定します。

送信レベル0（ゼロ）は、優先レベルをサポートしているNXファミリ（NX Dlinkなど）では、受信した場合の優先レベルを任意に割り付けることができます。詳細は、各機器に対応したNXサポートパッケージのマニュアルを参照してください。

また、ネットワークからメッセージを受信した場合は、ヘッダ内優先レベルを評価せず、到着順に処理します。

【制限事項】

NX Dlinkと接続する場合は、NX Dlink側で設定するメッセージ優先レベルに以下の制限事項がありますので注意してください。

NX DlinkからNXACP宛にロングメッセージ（2パケット以上で分割・再生が必要なメッセージ）を送信する場合は、1つの優先レベルを使用する必要があります。

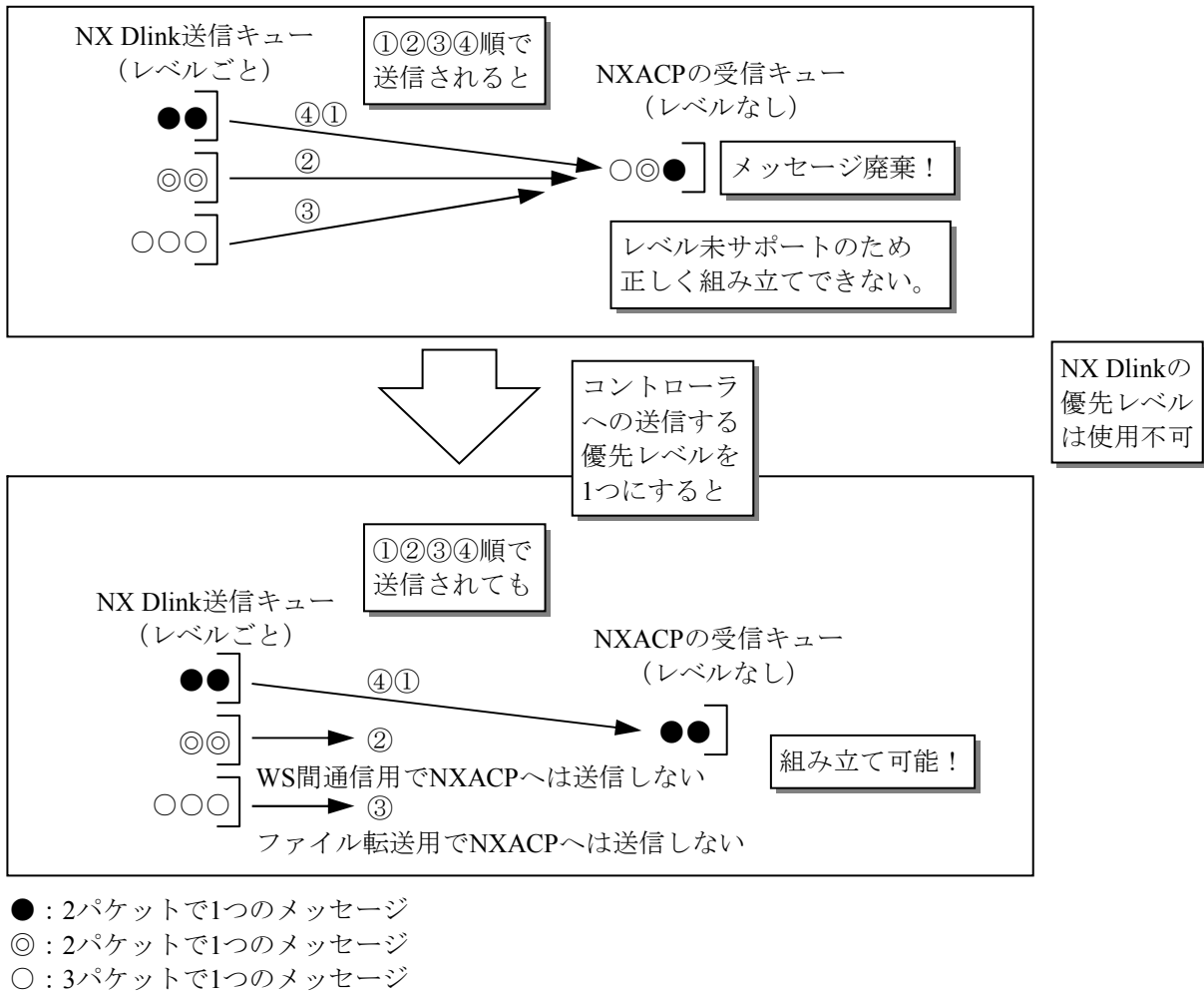


図2-7 優先レベル使用時の制限事項

1. 2 メッセージ送信機能

この節では、nx_put()マクロの送信機能について説明します。

ユーザタスクからnx_put()を発行する場合、送信先として「データフィールド番号 (DFN)」を指定する必要があります。

データフィールド番号に0以外を指定した場合、ネットワークに送信することを意味し、0を指定した場合は、自コントローラ内通信という特殊な意味を持っています。

- ネットワーク送信 (DF≠0を指定した場合)
送信者と受信者が異なるノードに存在することをネットワーク送信と呼びます。
- 自ノード内送信 (DF=0を指定した場合)
送信者と受信者が同じノード内に存在することを自ノード内送信と呼びます。

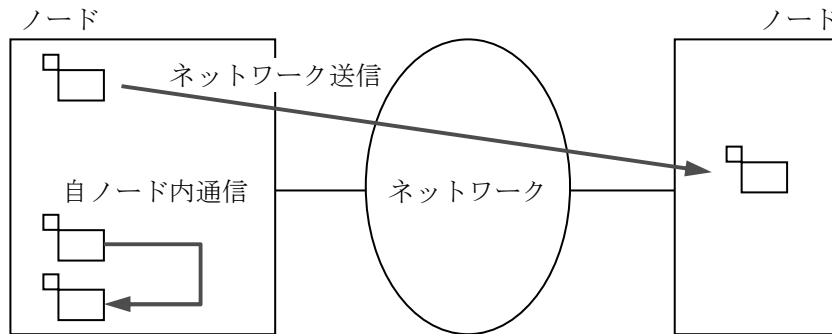


図2-8 データフィールド番号と送信範囲

1. 2. 1 ネットワーク送信

(1) 送信範囲の指定方法

ネットワークに接続された他ノードへメッセージ送信するには、データフィールド番号に0以外の番号を指定して、送信要求をする必要があります。

なお、送信に先立ち、構築情報でデータフィールド番号とネットワークアドレス (INA) との対応付けが必要です。

つまり、送信したいネットワークセグメントのアドレスとデータフィールド番号を論理的に対応させ、送信時はネットワークアドレスではなくデータフィールド番号を指定します。下記の例では、CTLからCPU11の接続されたネットワークにメッセージを送りたい場合、データフィールド番号は1を指定してnx_put()を発行します。

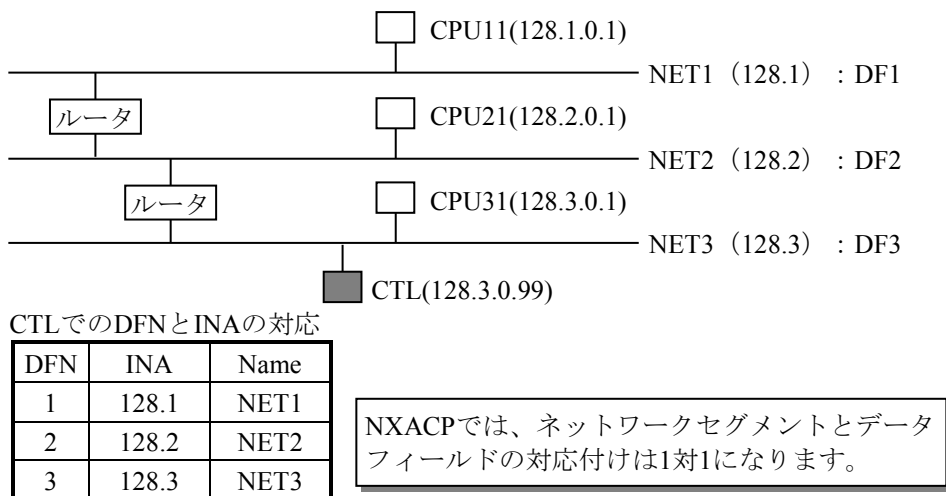


図2-9 データフィールド番号とネットワークセグメント

また、データフィールド番号の他に、マルチキャストグループ番号（MGN）とトランザクションコード（TCD）を指定する必要があります。

マルチキャストグループ番号も、以下のように、送信に先立ち、構築情報でオンライン/テストモードごとに、ポート番号と対応付けておく必要があります。

表 2-2 マルチキャストグループとポート

MGN	オンラインモード宛先ポート番号	テストモード宛先ポート番号
1	55001	57001
2	55002	57002
3	55003	57003
4	55004	57004

1つのマルチキャストグループには、モード別に2つのポートが対応する

NXACPでは、データフィールド番号で指定されたネットワークセグメントに、メッセージをブロードキャストします。ただし、このときの宛先ポート番号は、指定されたマルチキャストグループ番号に対応したポート番号です。

したがって、送信されたメッセージを受信できるのは、該当マルチキャストグループ番号に対応したポートを定義した（受信マルチキャストグループとして定義した）ノードだけとなります。ポートを定義していないノードでは、内蔵EtherはOSレベルでメッセージを廃棄します。

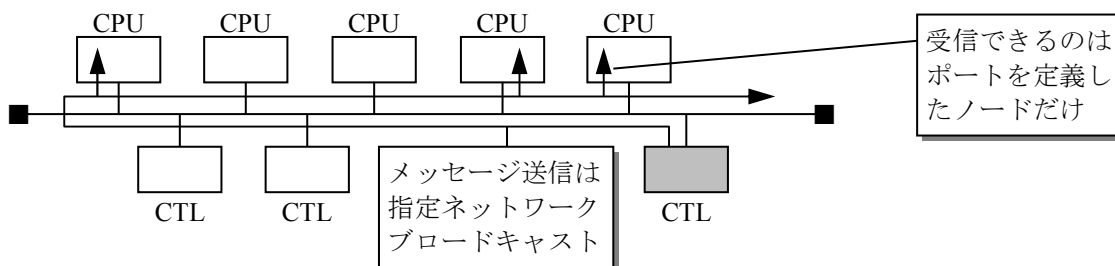


図 2-10 マルチキャストと選択受信

つまり、マルチキャストグループ番号を指定することでデータフィールドに接続されるノードの中で、送信したいノードにだけメッセージが送信されます。なお、マルチキャストグループ番号とポート番号の対応付けは、データフィールドに接続される全ノードで統一してください。

トランザクションコードについては、受信先ノードでメッセージを渡す相手（NXACPでは、ユーザタスク）を判定するIDです。NXACPでのトランザクションコードとユーザタスクの関係については、「1. 3 メッセージ受信機能」で説明します。

(2) リモートデータフィールドへの送信

ルータを経由して接続されたリモートデータフィールドへメッセージを送信する場合も、データフィールド番号を指定して送信要求します。このとき、リモートデータフィールドへの送信経路は、1経路だけ設定できます。経路を二重化したい場合は、後述する二重化LAN制御で経路を二重化してください。

図2-11において、ノードAからノードBにメッセージを送信する場合、データフィールド3経由で送信する経路とデータフィールド2経由で送信する経路がありますが、NXACPでは、どちらか一方をユーザの指定（構築情報）によって静的に管理します。したがって、複数の経路を設定したり、障害発生時、異常を検出して経路を切り替えることはできません。

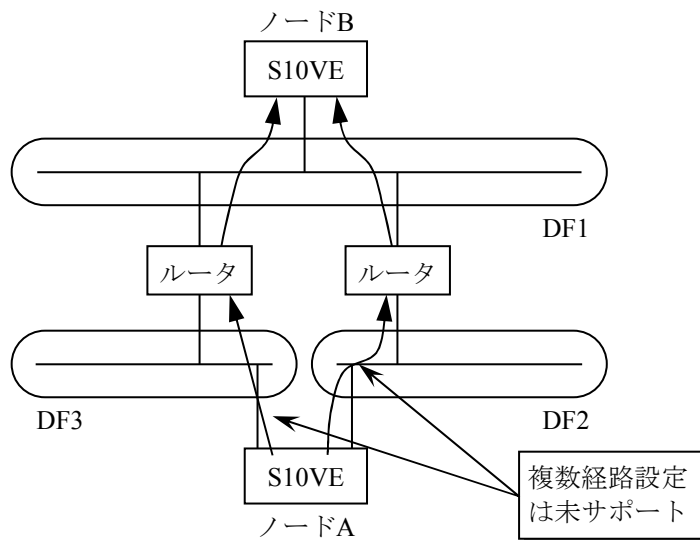


図2-11 リモートデータフィールド (1)

また、以下のような構築や送受信方法もできません。つまり、リモートデータフィールドは、1つのローカルデータフィールドに接続されている必要があります。

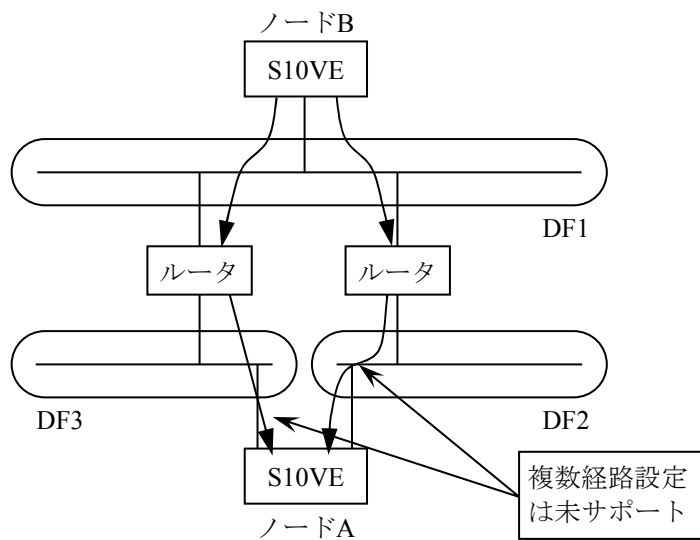


図2-12 リモートデータフィールド (2)

リモートデータフィールドは、1つのローカルデータフィールドに対応し、経路を1つとしてください。

通信の信頼性を向上させる場合は、ルータおよびネットワークを二重化することを推奨します。

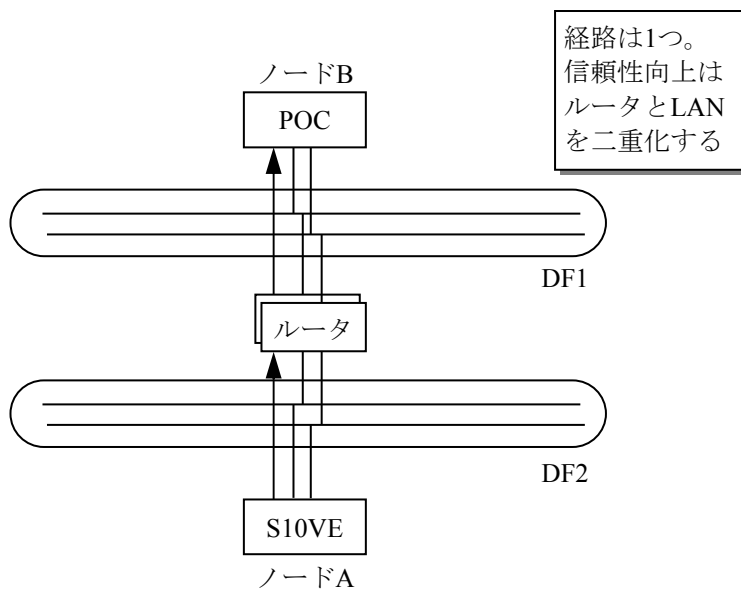


図2-13 リモートデータフィールド (3)

【留意事項】 リモートデータフィールド使用時の注意事項

リモートデータフィールドを使用する場合、システム設計時にリモート/ローカルデータフィールドのネットワークトラフィックや使用するルータの特性に注意してください。

(3) 自ノードへの折り返しについて

データフィールド番号≠0へのメッセージ送信要求は、自ノードへ折り返し処理はありません。送信したメッセージのマルチキャストグループ番号を、自ノードの受信マルチキャスト番号に定義している場合は、NXACPで廃棄します。NXACPの受信処理で送信元ノード番号を判定し、自ノードが送信元ノードのメッセージを廃棄します。

つまり、0以外のデータフィールド番号へのメッセージ送信要求は、他ノード送信専用です。

(4) 送信バッファ管理

送信バッファは、データフィールド単位に管理します。したがって、1つのデータフィールドの送信バッファがビジー状態になっても、他のデータフィールドへは影響がありません。構築時は、データフィールド単位にバッファ面数を指定してください。

(5) 送信権チェック処理

NXACPを使用してユーザタスクがメッセージを送信するためには、構築時、ユーザタスクごとに送信するトランザクションコードを定義することを推奨します。この定義情報に従い `nx_put()` では、送信元ユーザタスクと送信トランザクションコードが一致しているかチェックし、未定義トランザクションコードを送信しようとした場合、エラーとして処理を中断します。

この機能はオプション機能で定義も必須ではありませんが、トランザクションコードと送信ユーザタスクの対応を構築で定義し管理することを推奨します。

データフィールドごとに、1ユーザタスクが送信できるトランザクションコード数は、最大8TCDです。

表 2-3 トランザクションコードとユーザタスク対応

TCD	送信TN1	送信TN2	送信TN3	…	送信TN8
33	1	41	25		111
48	41	—	—		—
76	37	112	55		—
99	25	39	40		41

ユーザタスクと送信トランザクションコードの定義はデータフィールドごとに定義します。この定義は、省略することもできます。送信権チェック不要の場合、定義を省略してください。

送信権チェックの有無は、ユーザタスク単位に設定できます。送信権チェック実施と定義したユーザタスクは、送信先データフィールドすべての構築ファイルに送信トランザクションコードを定義してください。

(6) 特定ノードへの送信方法

NXACPでは、マルチキャスト送信だけをサポートしているため、宛先ノードを指定した送信はできません。しかし、マルチキャストグループ番号、トランザクションコードを論理ノードと対応させて定義することで特定ノード宛送信とすることができます。

ここでは、以下に示す2とおりの例を示します。

(a) 宛先ノード番号=マルチキャスト番号と定義する方法

マルチキャストグループごとに宛先ノードを対応するように定義することで、特定ノード宛送信を実現できます。

- ・各マルチキャストグループの受信ポート番号を各ノード専用に設定してください。二重化CPUの場合は、2つのノードに1つのマルチキャストグループを対応させてください。
- ・各ノード専用に定義したマルチキャストグループ番号を、データフィールド内の全ノードで統一してください。

表2-4 マルチキャストグループとノード対応

MGN	宛先ポート番号	対応ノード番号
1	55001	A
2	55002	B
3	55003	C
4	55004	D、E

全ノードで統一

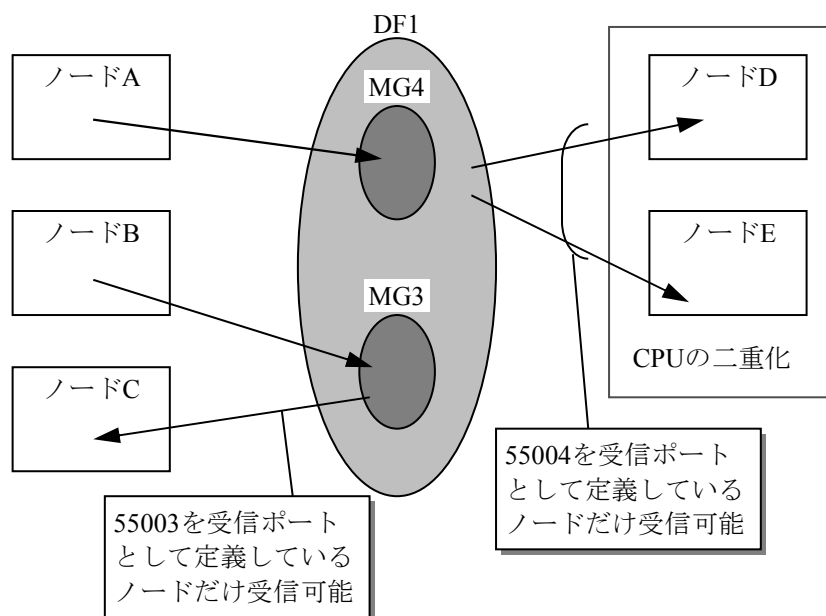


図2-14 特定ノード宛送信 (1)

NXACPでは、データフィールドごとに送信マルチキャストグループは最大128、受信マルチキャストグループは最大32グループまで定義できます。

(b) 宛先ノード番号=トランザクションコードと定義する方法

トランザクションコードごとに宛先ノードとするように定義することで、特定ノード宛送信が行えます。

- 各トランザクションコードを各ノード専用に設定してください。二重化CPUの場合は、2つのノードに1つのトランザクションコードを対応させてください。
- 各ノード専用に定義したトランザクションコードは、データフィールド内の全ノードで統一してください。

表2-5 トランザクションコードとノード対応

MGN	TCD	対応ノード番号
1	1~10	A
1	11~20	B
1	21~30	C
2	31~40	D、E

全ノードで統一

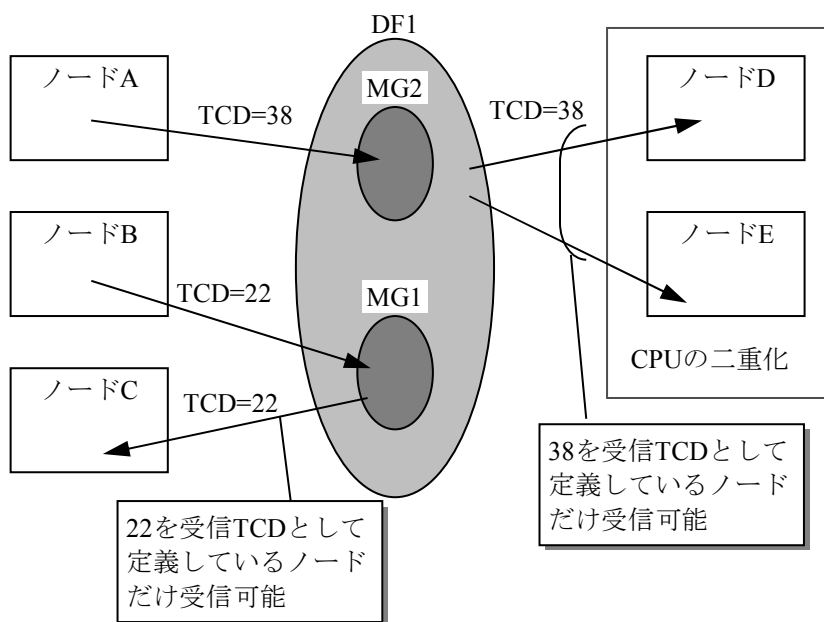


図2-15 特定ノード宛送信 (2)

(a) と (b) の方法を比較した場合、不要メッセージの廃棄を、(a) の場合ではハードウェアまたはOSレベルで実施し、(b) の場合はCPUのNXACPが実施します。したがって、CPU負荷を考慮した場合、(a) の使用方法を選択することを推奨します。

1. 2. 2 自ノード内送信

データフィールド番号0は、自ノード送信専用のデータフィールドです。

宛先データフィールド番号を0にしたメッセージを送信すると、自ノード内専用のバッファを使用しメッセージの受け渡しをします。このデータフィールドは、外部の他ノードとのメッセージ送受信はできません。

データフィールド番号0を指定して送信する場合には、マルチキャストグループ番号は意味を持ちませんので、0を指定してください。ユーザが0以外の番号を入力しても、NXACPIはマルチキャストグループ番号=0として処理します。

トランザクションコードは、自ノード専用データフィールドのトランザクションコードとなります。これは、トランザクションコードがデータフィールド単位のユニークであるためですが、受信タスクでは、他ノードからのメッセージと自ノード内メッセージを同じ手順で受信できるため、ネットワークワイドで機能とトランザクションコードを割り振ることを推奨します。

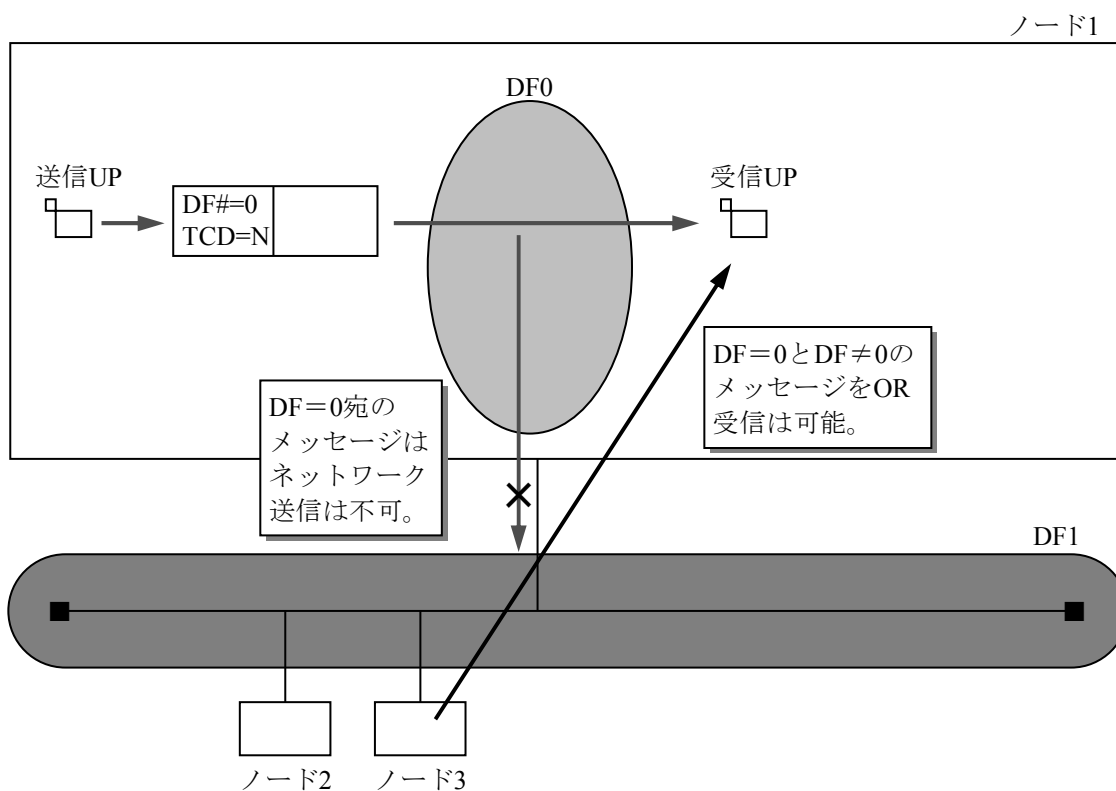


図2-16 自ノード内通信

送信バッファは、データフィールド単位の管理しますので、自ノード内通信でもデータフィールド番号=0専用の送信バッファを所有します。構築時、他データフィールド同様にバッファ面数を指定してください。なお、バッファ1ケースサイズも、データフィールド単位の管理となるため自ノード内専用サイズを設定できますが、他データフィールドと同じサイズで設定することを推奨します。

自ノード内通信では、送受信バッファは1つで管理（受信バッファとして管理）します。

送受信ごとに準備する必要はありませんので、構築情報定義時は、受信バッファだけ定義してください。

1. 3 メッセージ受信機能

この節では、`nx_get()`マクロの受信機能について説明します。

ユーザタスクから`nx_get()`を発行する場合、メッセージを受信するデータフィールドごとにトランザクションコードを定義しておく必要があります。この定義情報によって、NXACPからメッセージを受け取ることが可能となります。このとき、`nx_get()`を発行するユーザタスクでは、自ノード内通信やノード間通信を意識することなく受信することができます。

1. 3. 1 メッセージ受信

`nx_get()`でのデータ受信機能には、以下の特長があります。

- 1つのユーザタスクで、受信するデータフィールドやトランザクションコードを複数定義できます。
ただし、1回の受信要求で受信できるメッセージは1つだけです。なお、受信したメッセージの送信元アドレスなどの制御情報は参照できます。
- 1つの受信メッセージを複数のユーザタスクで共用受信することができます。したがって、1つのメッセージに複数の業務データを格納し、伝送効率を上げることができます。
- メッセージを受信する場合、データのオフセットとサイズを指定して受信できます。
したがって、1つのメッセージに複数の業務データが格納されていても、ユーザタスクごとに必要な情報だけを受信することができます。

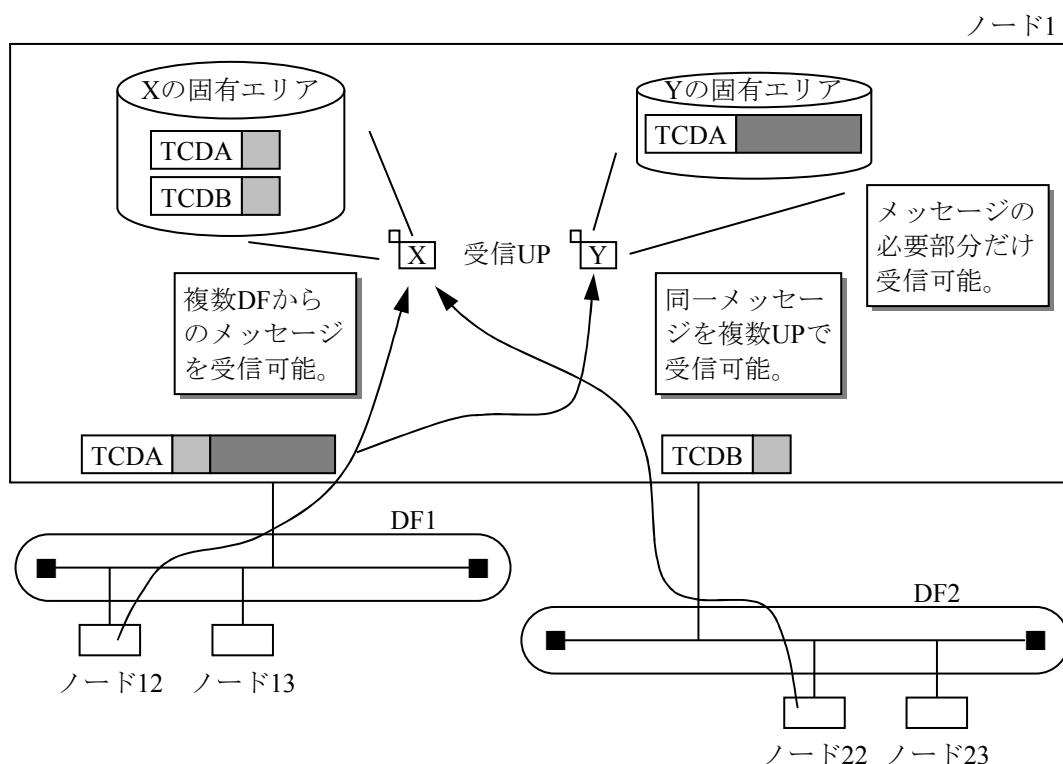


図 2-17 `nx_get()`の特長

(1) 定義情報と終了情報

NXACPを使用してユーザタスクがメッセージを受信するためには、構築時、ユーザタスクごとに受信するデータフィールド番号とトランザクションコードを定義する必要があります。

`nx_get()`のパラメータと分離しているのは、ユーザタスクと受信管理情報を分離させ、ユーザタスクの移植性を上げるためです。

データフィールドごとに、1ユーザタスクが受信できるトランザクションコード数は、最大8TCDで、同じTCDを最大8タスクで受信できます。また、1ユーザタスクで複数のデータフィールド番号にまたがって定義することもできます。ただし、1回の`nx_get()`発行で受信できるメッセージは、1メッセージだけです。

受信したメッセージは先着優先で処理されユーザタスクに渡されます。構築時指定したどのデータフィールドの、どのトランザクションコードかは、終了情報でユーザに通知します。

表2-6 トランザクションコードとユーザタスク対応

TCD	受信TN1	受信TN2	受信TN3	...	受信TN8
33	1	41	25		111
48	41	—	—		—
76	37	112	55		—
99	25	39	40		41

`nx_get()`マクロ終了情報

```
typedef struct {
    unsigned char dfn; /* 受信元データフィールド番号 */
    char fu1; /* 将来用 */
    unsigned short mgn; /* マルチキャストグループ番号 */
    unsigned short tcd; /* トランザクションコード */
    short fu2; /* 将来用 */
    unsigned long sa; /* 送信元アドレス */
    unsigned long rcvlen; /* 受信メッセージサイズ */
}nx_ginfo;
```

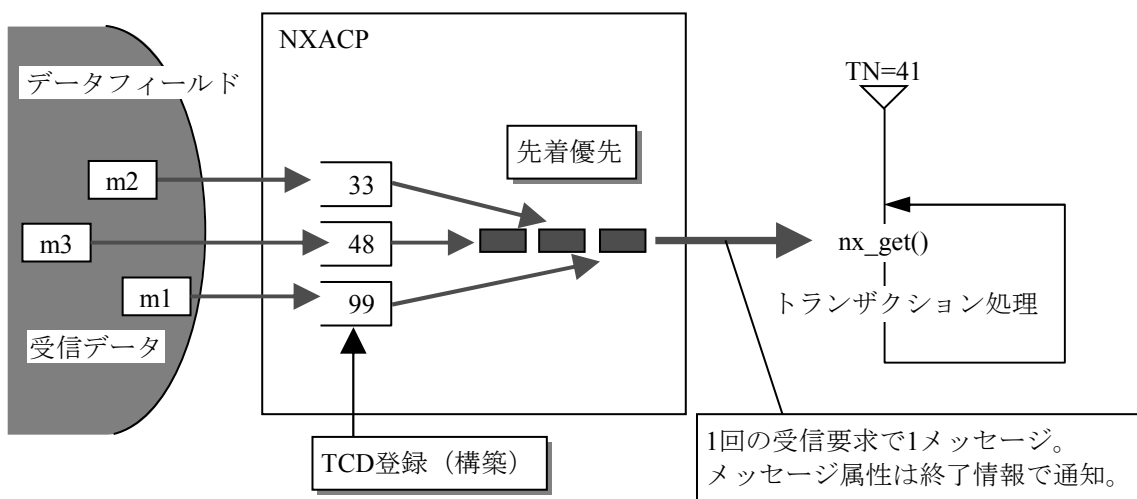


図2-18 受信処理の流れ

(2) トランザクションコード共用受信

1つのトランザクションコードメッセージを複数のユーザタスクで受信することができます。したがって、1つのメッセージに複数業務メッセージのデータを格納し、ネットワーク上での伝送効率を上げたり、ネットワークからの割り込み発生回数を少なくすることで、CPUの負荷を下げるすることができます。

また、ユーザタスクでも、複数業務メッセージを業務ごとに並列に実行できます。

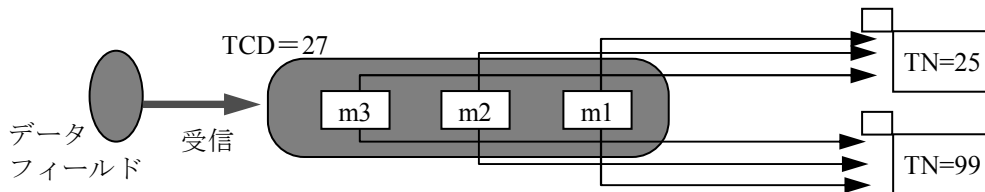


図2-19 共用受信

また、トランザクションコードを共用したユーザタスクでは、各タスクが必要なデータだけを切り出して受信できます。この機能によって、データコピー時間、使用メモリの低減を図ることが可能となります。

切り出す部分は、`nx_get()`マクロのパラメータで指定します。

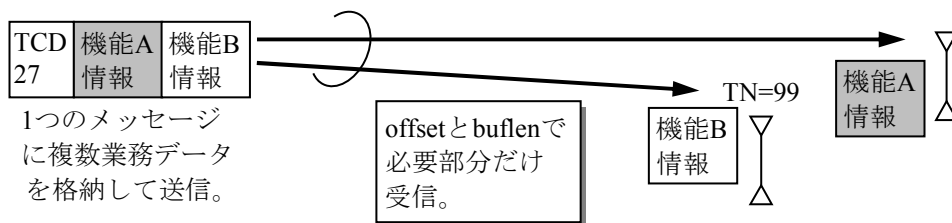
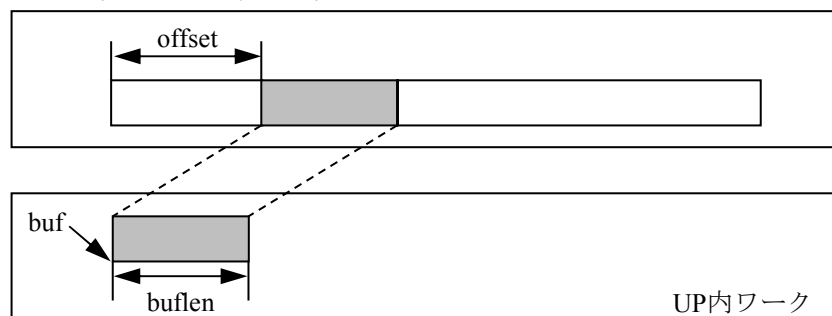


図2-20 切り出し受信

NXACP受信バッファ内の受信メッセージ



buf : UP内の受信ワークの先頭アドレス
 buflen : 受信すべきデータサイズ、かつ、受信ワークのサイズ
 offset : 切り出すメッセージの相対アドレス
 (メッセージ先頭を0とした相対アドレス)

図2-21 切り出し受信と使用メモリ

【留意事項】

複数のユーザタスクに受信されるメッセージは、受信すべきすべてのユーザタスクが受信完了したときにバッファ解放します。したがって、処理時間やプライオリティに差があるユーザタスクで同じトランザクションコードを受信する場合は、バッファ効率低下しいては受信バッファ満杯に伴うメッセージ廃棄を引き起こす可能性がありますので注意してください。

(3) 受信バッファ占有容量チェック

特定タスクの誤動作によって受信バッファがオーバーフローしないよう、トランザクションコードごとにバッファの使用面数を定義することができます。トランザクションコードごとに定義した面数を超過した場合、到着メッセージを廃棄します。この機能によって、特定のトランザクションコードで受信バッファを占有することを防止します。

トランザクションコードごとのバッファ使用可能面数は、構築時にユーザが指定します。廃棄されては困るトランザクションコードについては、バッファ面数を大きく確保してください。

1. 3. 2 受信タイムアウト監視

NXACPでは、ユーザタスクは、イベント待ち型（ループ型）の構成を推奨します。

`nx_get()`を発行するユーザタスクは、ループ型の構造を取り、ループの先頭でトランザクションの受信要求を行うことでオーバーヘッドを低減できるからです。

しかし、一定時間以内にメッセージが受信できない場合、ループから抜け出して処理しなければならない場合があります、これに対応できるよう`nx_get()`では、タイムアウト監視機能を備えています。

(1) タイムアウト監視時間の指定方法

`nx_get()`のパラメータ`time`に時間を指定してください。

指定可能な時間は、0～3600（秒：最大誤差±1）です。

ただし、0指定はメッセージ到着無限待ちを意味します。また、メッセージ未到着の場合で、メッセージの到着を待ちたくない場合、`time`には「-1」を指定してください。

(2) タイムアウト検知方法

`nx_get()`は、正常にメッセージを受信した場合、リターンコード=0でユーザタスクに制御を戻します。このとき、受信したデータサイズを`ginfo`の`rcvlen`に設定します。異常終了した場合は、リターンコード≠0でユーザタスクに制御を戻します。

メッセージ未受信の場合、NXACPでの処理としては正常と判断し、リターンコード=0でユーザタスクに制御を戻しますが、`ginfo`の`rcvlen`には「-1」を設定します。したがって、タイムアウト監視を有効として`nx_get()`を発行した場合（`time`>0）、リターンコードの判定後`ginfo`の`rcvlen`を参照し、タイムアウトによる待ち解除か、メッセージ受信による待ち解除かを判定して処理してください。また、ノンブロッキング受信（`time`に-1を指定時）を指定した場合も同様に、`ginfo`の`rcvlen`を参照して、メッセージ未受信による待ち解除か、メッセージ受信による待ち解除かを判定して処理してください。

1. 3. 3 マルチキャストグループの定義

ネットワークセグメントに対応するデータフィールド（≠0のDF）では、受信するマルチキャストグループ番号と受信ポート番号の対応付けを定義する必要があります。この定義がない場合、該当マルチキャストグループ宛のメッセージは受信できません。

この対応付けは、データフィールドに接続される全ノードで統一する必要があります。

表2-7 マルチキャストグループ番号と受信ポート番号

MGN	受信ポート番号
1	5501
11	5511

この対応付けを統一

送信時のマルチキャストグループ番号の定義にも宛先送信ポート番号を対応させますが、この宛先送信ポート番号と上記受信ポート番号は、同じマルチキャストグループ番号の場合、データフィールド内全ノードで統一する必要があります。

表2-8 マルチキャストグループ番号と送受信ポート番号

MGN	宛先送信ポート番号
1	5501
11	5511

この対応付けを統一

MGN	受信ポート番号	宛先送信ポート番号
1	5501	5501
11	5511	5511

受信ポートの定義は、MG1を受信したいノードで定義します。宛先送信ポートの定義は、MG1を送信したいノードで定義します。

この対応付けを統一

ユーザタスクは、メッセージ送信時、マルチキャストグループ番号を指定しますが、このときのメッセージは、NXACPによってユーザ指定マルチキャストグループ番号に対応した宛先送信ポート番号宛にブロードキャスト送信します。

他ノードでは、上記宛先送信ポート番号が受信ポートとして定義されていない場合、該当メッセージはハードウェアまたはOSで廃棄されます。宛先送信ポート番号と同じポート番号が受信ポートとして定義されているノードでだけメッセージを受信できます。

【留意事項】

1つのノードでマルチキャストグループの送受信ポートを両方定義した場合、他ノードからのメッセージは受信できますが、自ノードが送信したメッセージは受信できませんので、注意してください。

ポートとマルチキャストグループ番号の関係を送受信ごとに示すと図2-22のようになります。

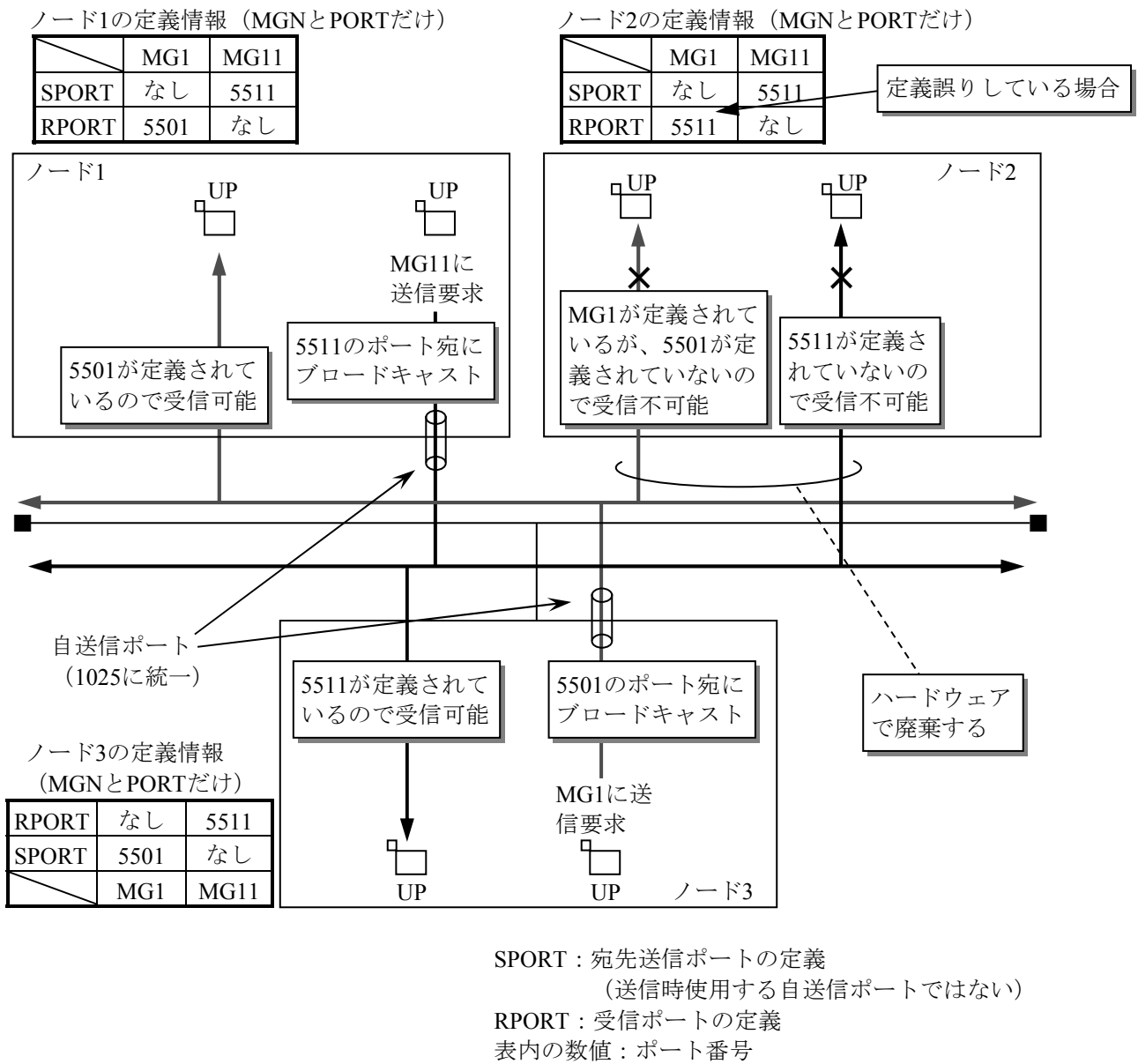


図2-22 宛先送信マルチキャストグループと受信マルチキャストグループ

1. 4 リモートデータフィールド制御機能

複数のLANをルータやノードで接続し、IP (Internet Protocol) のルーティング機能を利用することで大規模なネットワークにS10VEからアクセスすることができます。

1. 4. 1 システム接続形態

大規模ネットワーク構成の場合、あるノードから見ると、自ノードが直接接続されているLANとルータなどを介して間接的に接続されるLANに分けることができます。前者をローカルデータフィールドと呼び、後者をリモートデータフィールドと呼びます。

リモートデータフィールドは、必ず1つのローカルデータフィールドに接続されていなければなりません。

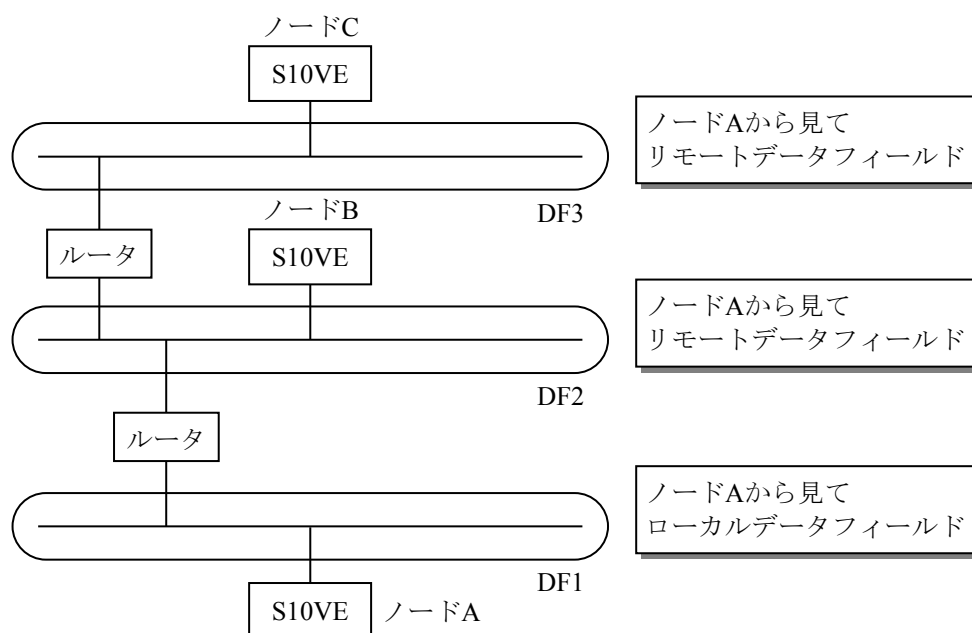


図 2-23 データフィールドタイプ

【留意事項】

リモートデータフィールドへの送信は、IP (Internet Protocol) のルーティング機能を使用した機能です。

ルーティング機能は、NXACPではなくRCTLNETで実現しています。

リモートデータフィールドは、送信経路として、必ず1つのローカルデータフィールドに対応しなければなりません。リモートデータフィールドとローカルデータフィールドを1対Nに対応させることはできません。

通信経路の信頼性を上げるために二重化経路を設定したい場合、「第3章 二重化LAN制御機能」を参照してください。

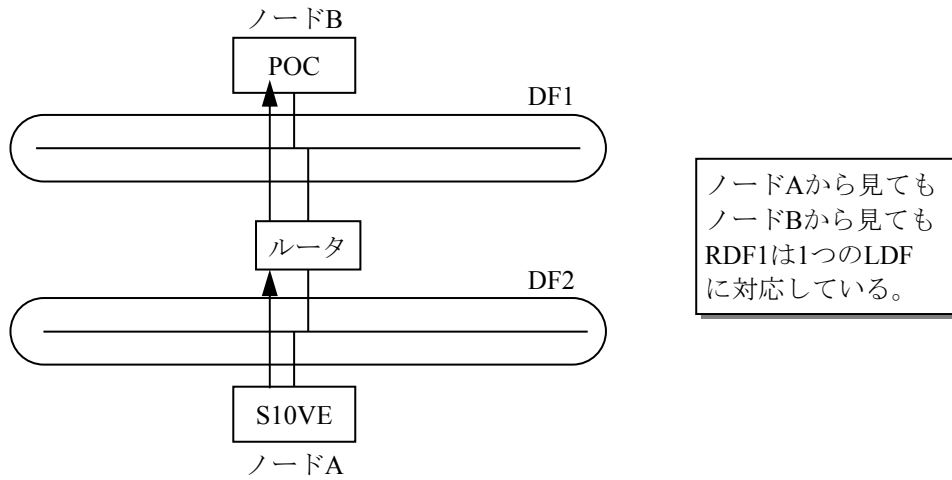


図2-24 リモートデータフィールド (1)

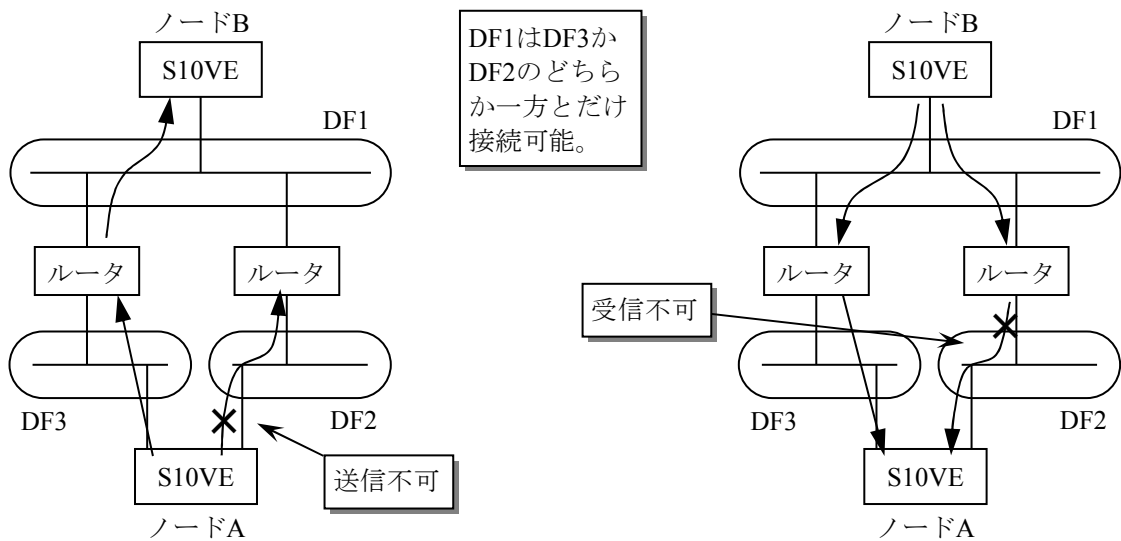


図2-25 リモートデータフィールド (2)

1. 4. 2 メッセージ送受信

(1) 通信の範囲

リモート/ローカルともに、そのデータフィールドで定義されているマルチキャストグループに対してメッセージを送信することは可能です。ただし、1回の送信要求で複数のデータフィールドや複数のマルチキャストグループへ同時に送信することはできません。

受信に関しては、各ノードは、ローカルデータフィールドで定義されたマルチキャストグループには加入できますが、リモートのマルチキャストグループには加入できません。したがって、各ノードのローカルデータフィールドに送信されるメッセージは受信できますが、リモートデータフィールドに送信されたメッセージは受信できません。

(2) メッセージの送受信方法

メッセージを送信する場合、宛先データフィールドがリモート/ローカルにかかわらず、宛先データフィールド番号とマルチキャストグループ番号、トランザクションコードを指定して送信要求 (`nx_put()`) を発行します。この場合、指定した宛先データフィールドでだけメッセージを受信することができます。

ルータをいくつか介したリモートデータフィールドへ送信した場合でも、宛先データフィールド以外（途中に存在するデータフィールドやローカルデータフィールド）では受信できません。

一方、受信メッセージは、送信元ノードがリモート/ローカルにかかわらず、メッセージが送信されたローカルデータフィールドの受信キューに接続されます。したがって、メッセージを受信する場合は、ローカルデータフィールド番号とトランザクションコードを定義して受信要求 (`nx_get()`) を発行する必要があります。なお、この場合、受信できるメッセージはローカルデータフィールド宛に送信されたメッセージだけです。

なお、リモートデータフィールドへの送受信サービスは、リモートデータフィールドが属するローカルデータフィールドの処理プログラムが対応します。したがって、ローカルデータフィールドが動作していない場合、リモートデータフィールドへの送信も停止します。

リモートデータフィールドとメッセージ送受信の関係をまとめると図2-26のようになります。

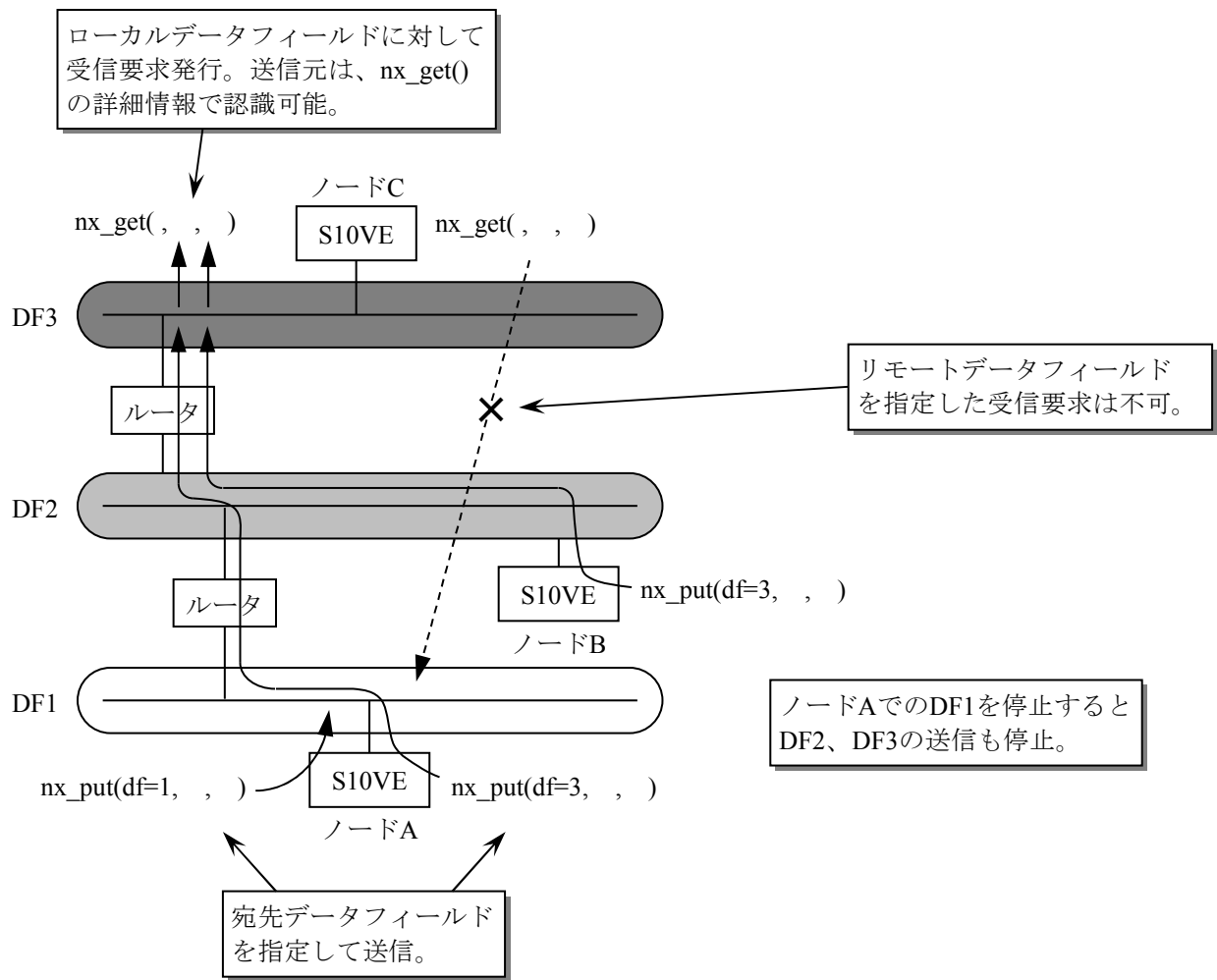


図2-26 リモートデータフィールドのメッセージ送受信

リモートデータフィールドの運用は、ローカルデータフィールドの動作状態に影響されません。運用の詳細は、「第6章 運用管理機能」を参照してください。

1.5 バッファ管理

送受信バッファは、データフィールドごとに送受信に分けて管理しますので、ローカル/リモートデータフィールドで送受信ごとに定義してください。ただし、自ノード内通信用データフィールド(番号=0)では受信バッファだけ指定してください。

表2-9 データフィールドタイプとバッファ定義

	ローカルデータフィールド (自ノード内専用)	ローカルデータフィールド (自ノード内除く)	リモートデータフィールド
送信バッファ	定義不要	定義要	定義要
受信バッファ	定義要	定義要	定義要

バッファ管理として、バッファオーバーフロー/HIGH-WATERアラーム/LOW-WATERアラームの検知機能とEAS通知機能をサポートしています。バッファ管理に関しては「5.1 障害通知機能」を参照してください。

第2章 データフィールド管理機能

NXACPでは、データフィールドの状態管理機能として下記の機能があります。

- ・自ノードの生存信号送信
- ・他ノードの生存状態監視および状態変化通知機能

NXを搭載した各ノードは、一定間隔で各ノードが直接接続するローカルデータフィールドに対して「生存信号」をブロードキャスト送信します。他ノードでは、この生存信号を受信することによってデータフィールドに接続した各ノードの生死状態を監視することができます。

NXプロトコルの生存信号には、以下の3つのモードがあります。

- ・生報告（通常モード）
- ・SHUTDOWN予告報告（計画的な機器停止）
- ・メンテナンス予告報告（メンテナンスのための一時的な機器停止）

NXACPでは、NXプロトコル生存信号の3つのモードに対し、生報告とSHUTDOWN予告報告をサポートします。メンテナンス予告報告は、SHUTDOWN予告報告と同等として扱います。

2. 1 生存信号送信機能

NXACPは、イニシャル要求を受け付ける（`nx_dfup()`が発行される）と「生」報告の生存信号を、構築時指定された一定間隔で送信します。

また、停止要求を受け付ける（`nx_dfdown()`または`nx_quit()`が発行される）と、「SHUTDOWN予告」報告の生存信号を送信します。この「SHUTDOWN予告」によって、他ノードでは、計画停止なのか異常による停止なのかを判断することができます。

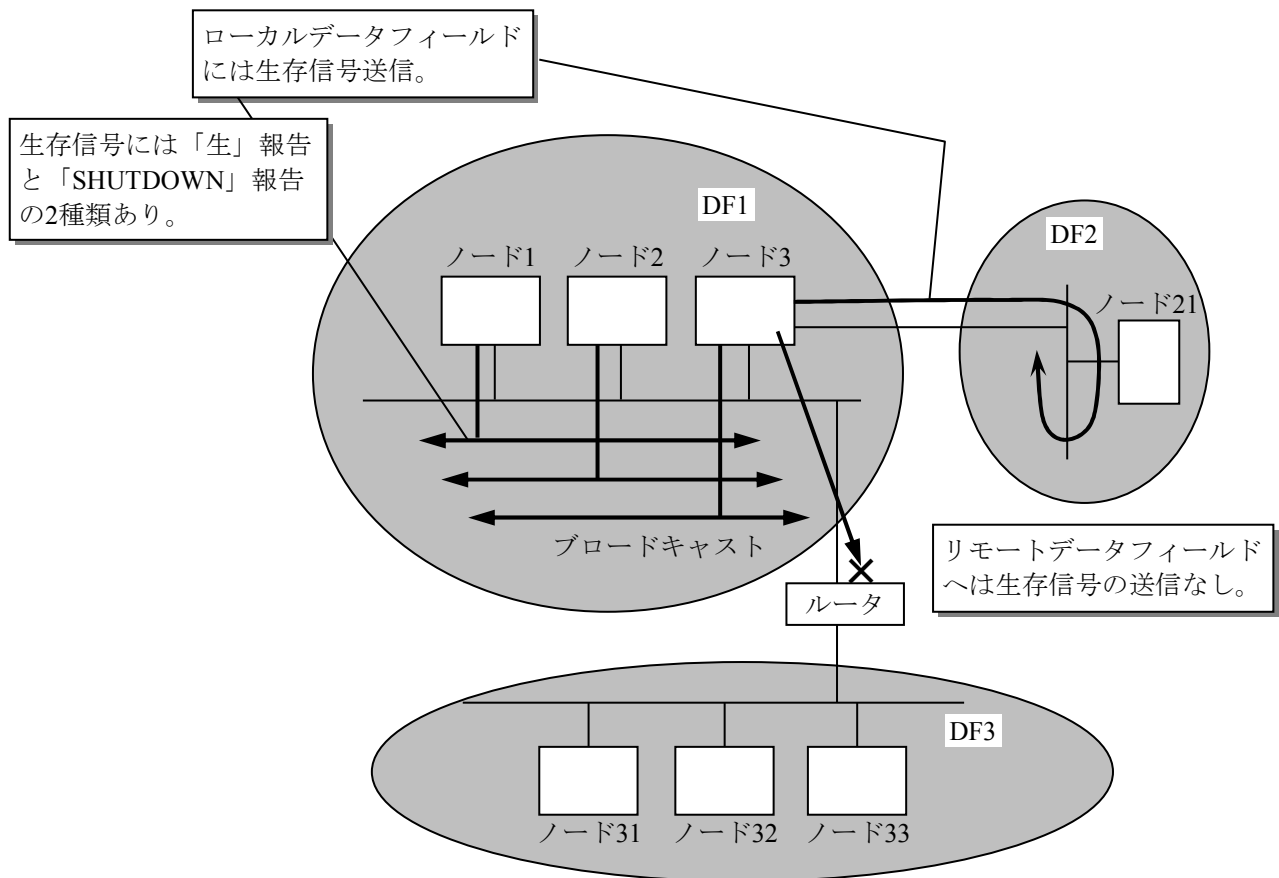


図 2-27 生存信号

生存信号によって、データフィールド単位に他ノードのデータフィールドへの接続状態を監視することができます。ここで言う監視とは、システム内で特定ノードが動作しているか否かではなく、業務のベースとなるデータフィールドから見て、各ノードがデータフィールドに接続され他ノードと協調しあいながら稼働できるか否かの監視です。

例えば、図2-27に示す構成でノード3は、データフィールド1と2に接続されています。この構成でデータフィールド1とノード3の接点の通信機器（ケーブル、イーサネットモジュールなど）が故障した場合、データフィールド1に接続する他ノードは、ノード3が「生」から「死」状態に変化したと認識し、その旨ユーザに通知します。しかし、データフィールド2では、ノード3は他ノードからは「生」状態のままと認識され続けます。

NXACPの送信する生存信号のタイプを表2-10に纏めます。

表2-10 生存信号タイプ

報告タイプ	生報告	SHUTDOWN予告
送信開始タイミング	UPからのnx_dfup()要求	UPからのnx_dfdown()/nx_quit()要求
送信回数	停止要求まで無限	3回
送信周期	ユーザ指定（秒単位）	nx_dfdown()/nx_quit()発行直後と1秒後、2秒後

nxdfup()/nx_dfdown()/nx_quit()（詳細説明は「第6章 運用管理機能」参照）は、NXACPのデータフィールドごとの立ち上げ/停止を要求するマクロです。

つまり、データフィールド単位に立ち上げ要求をnx_dfup()で発行すると同時に「生報告」の生存信号を一定周期で送信し始めます。

また、停止要求をnx_dfdown()またはnx_quit()で発行すると「SHUTDOWN予告」の生存信号を、マクロ発行直後と1秒後、2秒後の3回だけ送信します。

「生報告」の送信周期は、ユーザが構築時に秒単位に指定できますが、ユーザが構築時に定義する情報としては他にも表2-11に示すものがあります。それぞれの情報は、生存信号に付加してブロードキャストされNX Dlinkのdfstatコマンドを使用して参照できます。

表2-11 生存信号付加情報

ユーザ定義情報	説明
ノード番号	データフィールド単位に付ける番号
ノード名称	データフィールド単位に付ける名称
生存信号タイムアウト監視時間	生存信号が未到達となってから状態が「死」に変化したと認識するまでの時間（詳細は、「2.2 生存状態監視と状態変化通知機能」参照）。
IPアドレス	論理ノード単位のINA

生存信号を送出する場合のノードモードは、nx_dfup()のパラメータで指定されたモードとなります。

2. 2 生存状態監視と状態変化通知機能

生存状態監視とは、他ノードから一定周期で送信される生存信号の到達を、ノード単位に監視する機能で、生存信号が未到達になった場合や「生」報告から「SHUTDOWN予告（メンテナンス予告）」報告に変化したかどうかを監視します。

状態変化通知とは、他ノードの状態に変化があった場合、変化した内容とノード番号をユーザに通知する機能です。

生存状態監視は、構築時にユーザが使用／未使用を指定する必要があります。状態変化通知は、生存状態監視を使用する場合だけ使用できます。

【留意事項】

生存状態監視機能を使用する場合、監視するデータフィールドに接続するノード数や他ノードの生存信号送出周期によって、受信側コントローラの負荷が著しく上がる場合がありますので注意が必要です。通常はWSかPCで監視するか、またはデータフィールド監視専用のコントローラ

（WSやPCがデータフィールド内に存在しない場合）で監視することを推奨します。

特に生存信号送出周期は、データフィールドに接続する全ノードで一斉に送出された場合、内蔵Etherでは64ノードまでの処理が限界です。これ以上のノード台数をS10VEで生存監視すると、ノードは生存しているにもかかわらず、生存信号を受信できないために「死」と誤判断する場合があります。

64ノード以上の生存監視をする場合、WSまたはPCで状態監視してください。

(1) 状態管理

NXACPが認識するノード状態を以下に示します。

生（接続）：生存信号（生報告）が定期的に届いている状態です。

該ノードは、データフィールドに接続され、健全とみなします。

死（切離）：生存信号が一定時間内に一度も届いていない状態です。

または、SHUTDOWN予告（メンテナンス予告）を受信した状態です。

該ノードは、該データフィールドから切り離され、死とみなします。

上記の一定時間とは、ユーザが構築情報で指定した生存信号タイムアウト監視時間を指します。生存信号タイムアウト監視時間とは、一定周期で送信されてくるはずの生存信号が、どの程度到達しなかった場合に「死」と認識するかを示す時間です。したがって、生存信号タイムアウト監視時間と生存信号送信周期時間には以下の関係があります。

$$\text{生存信号タイムアウト監視時間} > \text{生存信号送信周期時間}$$

ネットワークのふらつきや負荷などによって、生存信号の packets が消滅する場合がありますので、生存信号タイムアウト監視時間は生存信号が一度抜けただけで死と認識しない時間で設定することを推奨します。

(2) 変化通知

NXACPでは、各ノードの状態に変化を認識した場合、ユーザに通知します。

ただし、SHUTDOWN予告とメンテナンス予告の生存信号は同等に扱い、ユーザへの通知は「SHUTDOWN予告での死」として報告します。

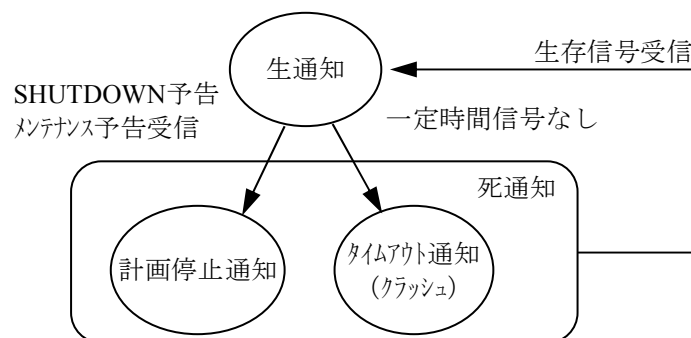


図 2-28 変化通知

状態変化が発生した場合、ユーザに通知する変化通知について、そのタイムシーケンスを以下に示します。

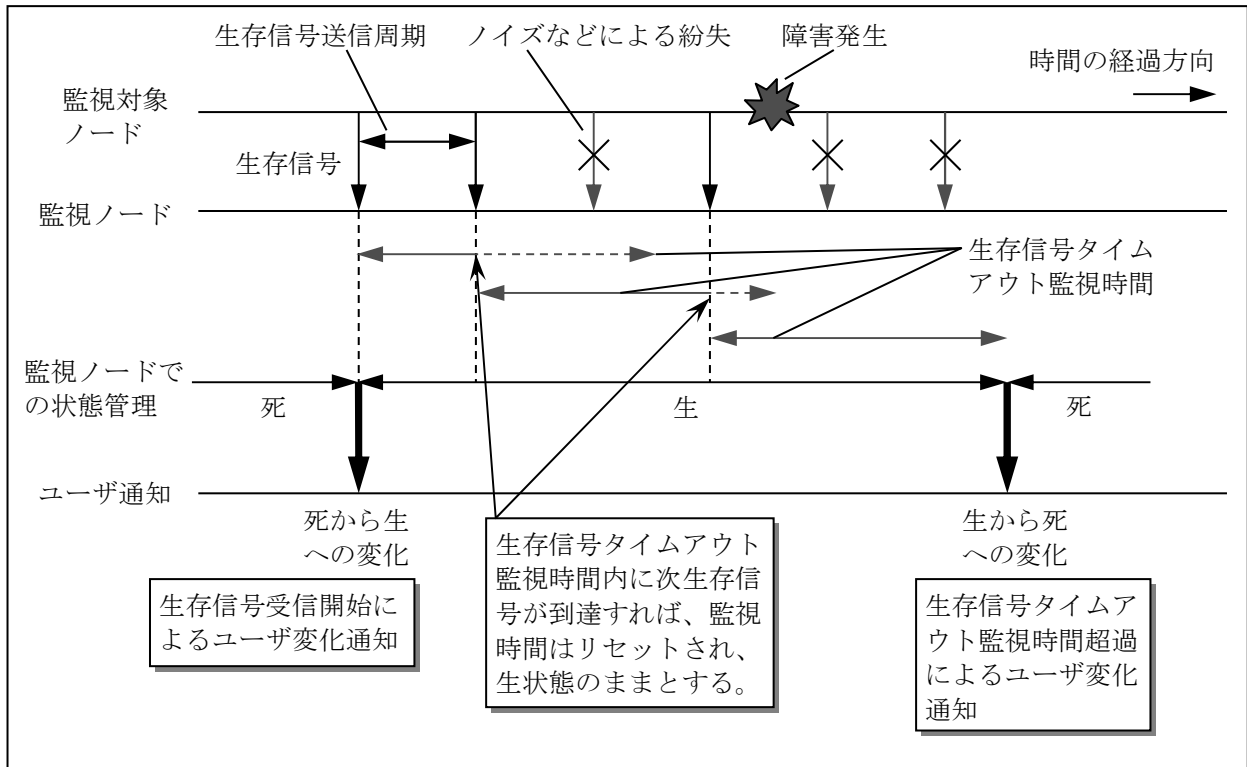


図2-29 変化通知のタイムシーケンス (タイムアウト)

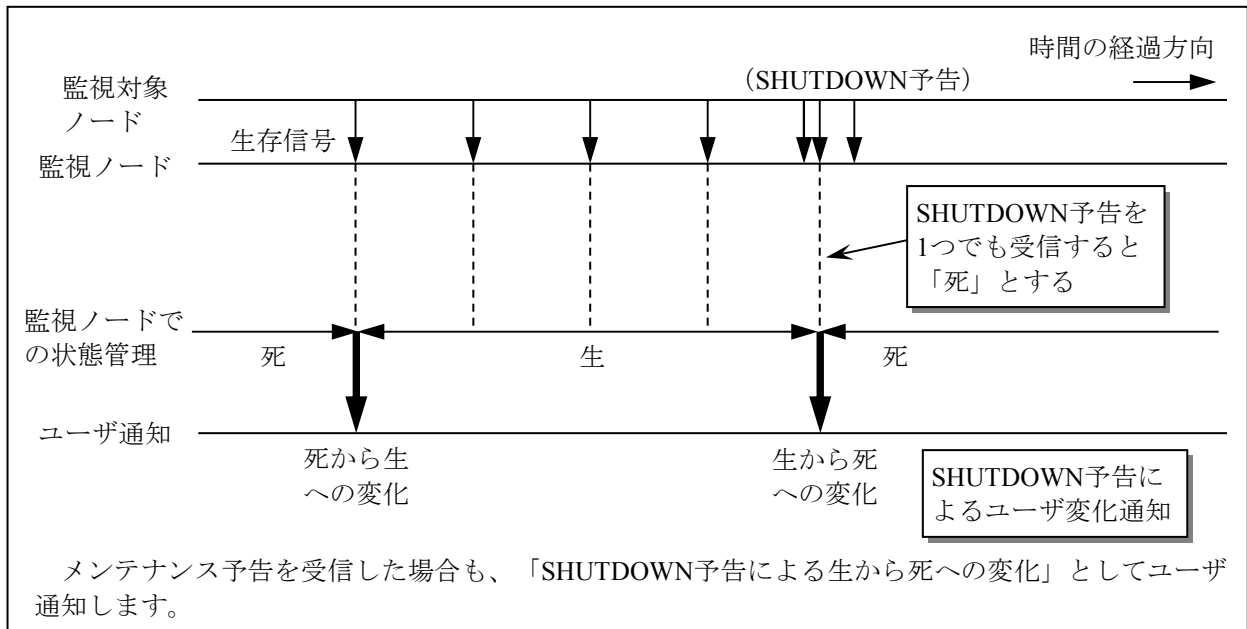


図2-30 変化通知のタイムシーケンス (SHUTDOWN警告)

(3) ユーザ通知条件

ユーザへの状態変化通知は、すべてIRSUB (#332) にリンクします。情報詳細は「付録C ノード状態変化通知フォーマット」を参照してください。ここでは、通知のタイミングとその性質について記載します。

ノード状態変化通知は、状態監視機能を使用しているノードのユーザ (UP) だけが検知可能でかつ、自ノードが属するローカルデータフィールド内のノードに限定されます。

生状態への変化は、`nx_dfup()`発行以降、他ノードの生存信号が到達した場合にユーザ通知しますが、死状態への変化は、一度も生報告の生存信号を受信していないノードから、SHUTDOWN予告やメンテナンス予告を受信した場合は、ユーザ通知はしません。死状態への変化は、`nx_dfup()`発行以降、一度でも生報告の生存信号を受信したノードに対して検知します。検知するタイミングは、生存信号タイムアウト監視時間超過時とSHUTDOWN/メンテナンス予告の生存信号を受信した場合です。

【留意事項】

- ・ 自ノードの扱い

自ノードの状態遷移については、`nx_dfup()`発行時、`nx_dfup()`処理完了時に一度だけ、「生」に変化したことをユーザに通知します。ただし、`nx_dfdwn()/nx_quit()`発行による「死」変化通知はありませんので注意してください。

- ・ 停止要求の扱い

`nx_dfdwn()/nx_quit()`発行後は、該当データフィールド処理を停止しますので、「死」変化通知は、自ノードも含め全ノードについて報告しません。`nx_dfdwn()`の処理完了をもって全ノード「死」と認識してください。

- ・ 変化通知は、生存信号を受信するモード (構築指定) であることと、NXACPの起動 (`nx_init()`) 前にIRSUB (#332) が登録されている必要があります。

第3章 二重化LAN制御機能

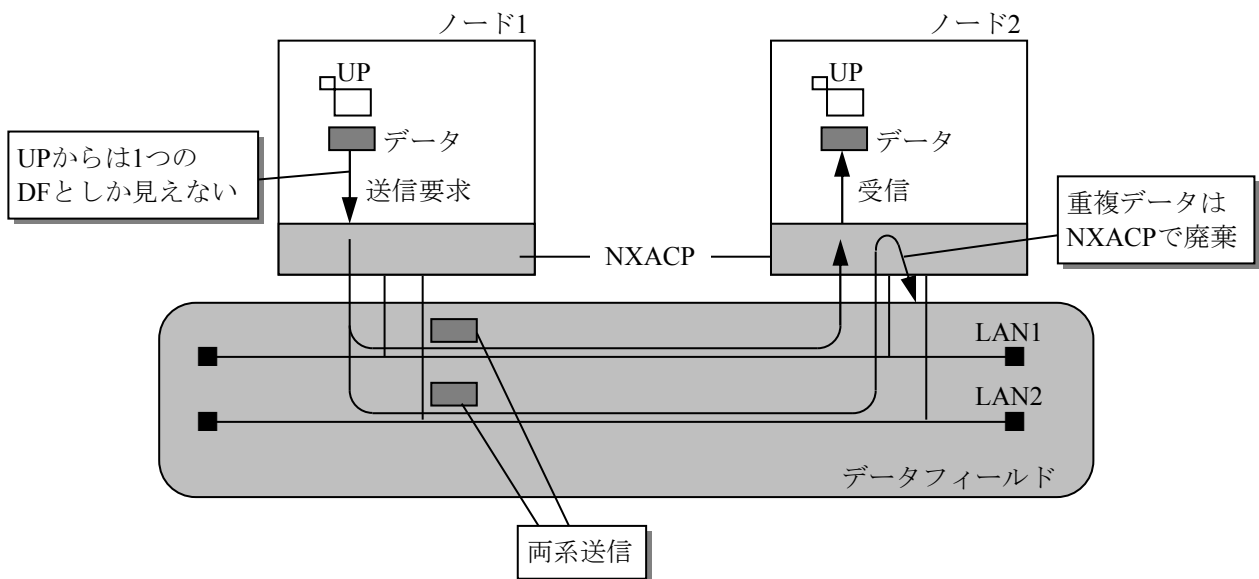
イーサネットの各ネットワークにおいては、ネットワークの特性上、コントローラの電源OFF/ONや、ネットワーク負荷の増加に伴いネットワーク上でパケットが消滅する場合があります。このような場合は、送信をリトライする必要があります。しかし、このリトライ処理によってネットワーク/CPUの負荷を上げ、結果としてシステム全体を高負荷状態としてしまう場合があります。

NXのマルチキャスト通信における二重化LAN制御は、データ送信時の欠損を低減し、通信の信頼性向上を目的としています。

3. 1 二重化LANでのメッセージ送受信

二重化LAN制御機能の運用上の特長として、以下の点が挙げられます。

- ・NXACPを利用するユーザタスクからは、物理的な2つのLANは、1つのデータフィールドとして見えるためLAN1、LAN2を意識する必要がありません。
- ・NXACPでサポートする二重化通信方式は、両LANに送信し、両LANから受信します。同じメッセージの重複受信は、NXACPで廃棄します。マルチキャストグループ番号に対応する送信宛先ポート番号や受信ポート番号は、同じポート番号を使用します。したがって、片系LANに障害が発生した場合でもLANの切り替え（構成制御）は不要です。



生存信号メッセージも両系送信されます。したがって、片系LANに障害が発生しても、メッセージ欠損にはなりません。また、片系LAN障害時の構成制御も不要です。

図2-31 二重化LANでのメッセージ送受信

【留意事項】

二重化対象のLANは、同じ種類のネットワークに限ります。これは、NXACP宛に送信する二重化LAN経路で、ネットワーク伝送路遅延による到着時間に著しく差があった場合、NXACPでは、初めて受信したメッセージを重複データと誤判断して廃棄してしまう場合があるからです。

二重化LANを使用して、リモートデータフィールドと接続する場合、ローカル／リモートの両データフィールドを二重化LANにしてください。

- ローカル／リモートデータフィールド共二重化LAN構成の場合、それぞれに対応するLANに送信します。なお、システム構築時は、LANとルータのルーティングアドレス対応に、不整合が発生しないよう注意してください。

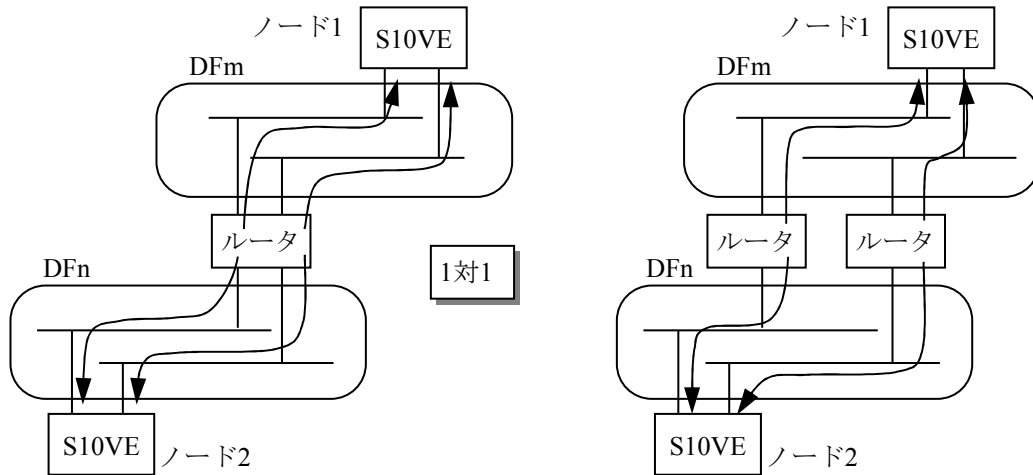


図2-32 二重化LANとリモートデータフィールド (1)

二重化した場合も、リモートデータフィールドとローカルデータフィールドでのLANの対応は1対1にする必要があります。経路設定も1つです。

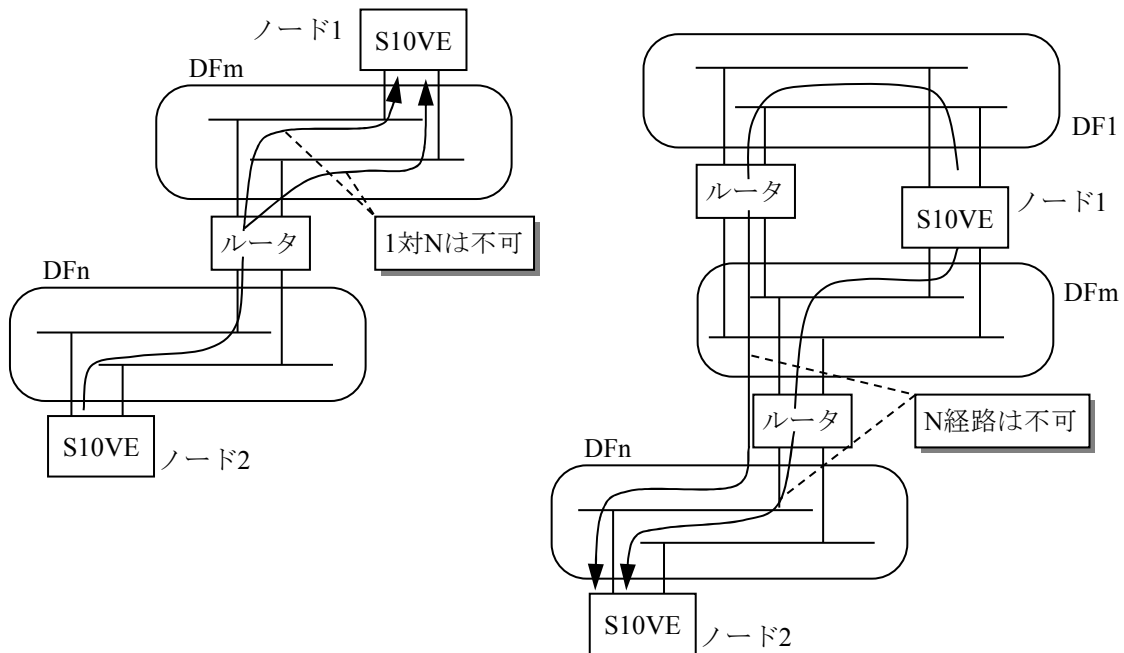


図2-33 二重化LANとリモートデータフィールド (2)

ローカルデータフィールドからリモートデータフィールドの片方だけ二重化LANでも、構成は可能ですが、以下の制限事項があります。

- 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でLAN構成数が異なる場合、宛先データフィールドの接続分送信します。ローカルデータフィールドが一重化LANの場合でも、リモートデータフィールドが二重化LANの場合、2回送信します。

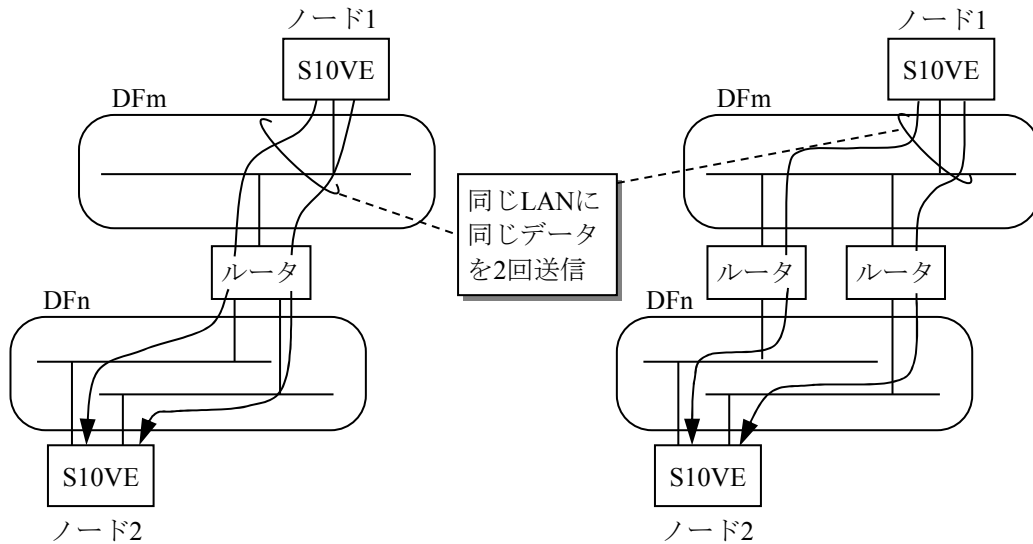


図2-34 二重化LANとリモートデータフィールド (3)

ローカルデータフィールドが二重化LANで、リモートデータフィールドが一重化LANの場合、対応するLANは、ローカルデータフィールドの構築情報で先に定義されたLAN（データフィールド定義情報のUNO1に定義したLAN）を意味します。

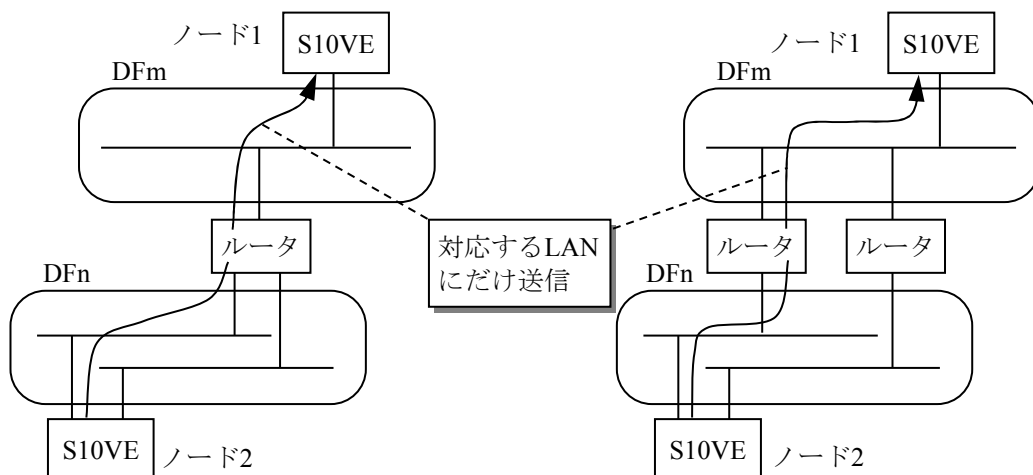


図2-35 二重化LANとリモートデータフィールド (4)

3. 2 二重化LANでの生存信号送受信

生存信号は、データフィールドを構成するLANが二重化されている場合、それぞれのLANに対して送信します。また、他ノードの生存監視もLANごとに行いますので、この機能によって、各ノードに障害が発生しているのか、ネットワークに障害が発生しているのかを判断することができます。

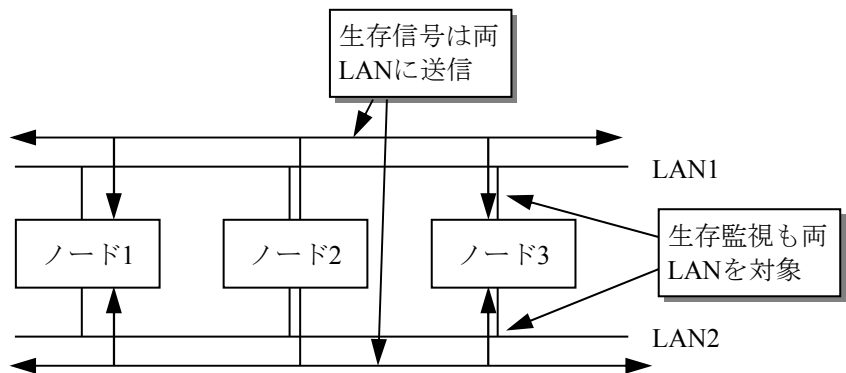


図 2-36 二重化LANと生存信号

(1) 生存信号送信

生存信号を二重化LANに送信する場合、一重化LAN時送信する生存信号と同じフォーマットで同じデータのをLAN1、LAN2に送信します。

(2) 状態監視と変化通知

状態監視と変化通知は、LAN単位に実施します。

つまり、LAN1から生存信号を受信していても、LAN2から生存信号が未到達となり、未到達時間が生存信号タイムアウト時間を超過した場合、そのタイミングでユーザに、「LAN2側のノードNが生から死に変化」したとして通知します。また、「死から生に変化」した場合の通知も、LANごとに報告します。

したがって、ノードに変化が発生した場合、通常はLAN1/LAN2の両方から2回、ユーザに状態変化が報告されますが、これはパケット消滅などによって常に保証される訳ではありませんので注意してください。

状態変化通知のフォーマット詳細は「付録C ノード状態変化通知フォーマット」を参照してください。

第4章 テスト機能

テスト機能とは、ユーザプログラムを容易に種々のテスト形態でテストするための支援機能です。システムを段階的に拡張していく上で、既存のオンラインで動作しているコンポーネントを変更したり、テスト専用システムを作成することなく、ノード単位でのテストを可能としています。

NXACPでは、既存のコンポーネントに影響を与えることなくテストを可能とするために、ノードとメッセージのそれぞれにモードを付加し、このモードの組み合わせによって、ノード単位でのメッセージ入出力制御を行います。つまり、ノード単位に入出力するメッセージを限定することでシステムの一部のコンポーネントだけを、論理的にシステムとは隔離して運用できます。

なお、この章で記載する機能は、マルチキャスト通信を対象にしています。

4.1 メッセージ入出力制御

メッセージ入出力制御方式は、ユーザが構築で指定します。

メッセージ入出力制御の組み合わせで、ノードモードとメッセージモードの可能な組み合わせは以下のとおりです。

表 2-12 ノードモードとメッセージモード

ノードモード	送信するメッセージモード	受信できるメッセージモード
オンライン	オンライン	オンライン
テスト	テスト	テスト
テスト	テスト	オンライン

オンラインノードでは、送受信メッセージはオンラインメッセージだけとなり、受信可能メッセージもオンラインメッセージだけとなります。なお、テストメッセージ用のポートはイニシャライズしないため、OSレベルで廃棄されますので、テストメッセージによるアプリケーションへの擾乱はありません。

テストノードでは、送信メッセージはテストメッセージだけとなりますが、受信可能なメッセージは以下のどちらかを選択することができます。

- ・テストだけ
- ・オンラインだけ

どちらの入力方式を選択した場合でも、受信不要のモード用ポートをイニシャライズしないため、不要モードメッセージはOSレベルで廃棄されますので、受信不要のメッセージによる擾乱がアプリケーションに影響することはありません。

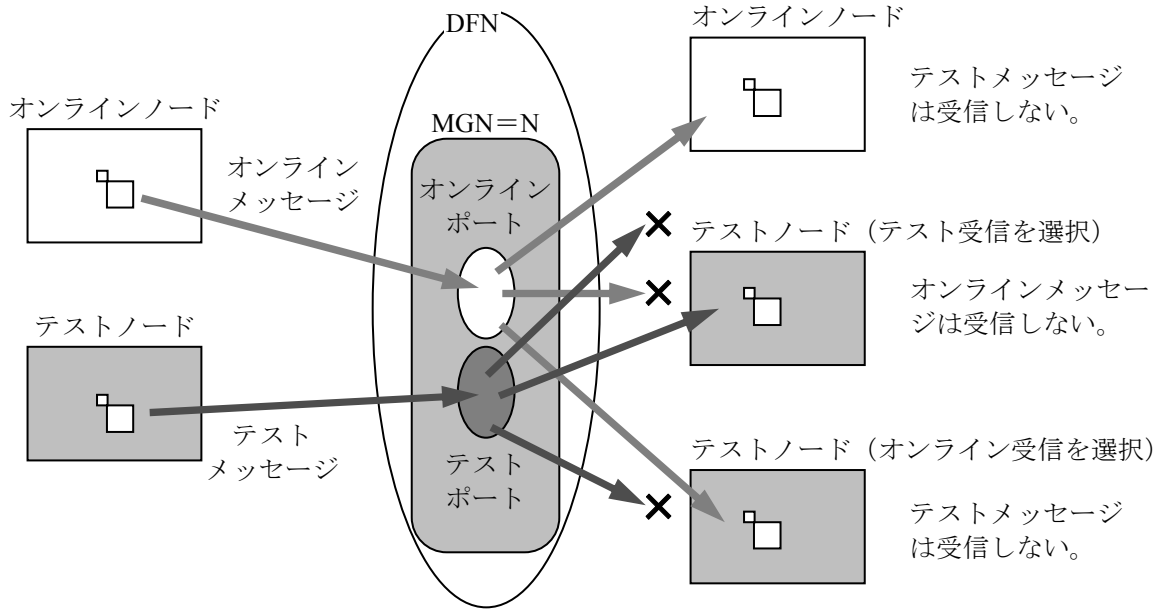


図 2-37 メッセージ入出力制御

図 2-37で、オンラインノードがNのマルチキャストグループにオンラインメッセージを送信する場合、NXACPは、Nのマルチキャストグループに対応するオンライン用ポート番号を送信宛先ポートとしてメッセージを送信します。したがって、ユーザはオンラインノードを構築時指定した場合、Nのマルチキャストグループに対応するオンライン用ポート番号を送信用として定義する必要があります。

また、Nのマルチキャストグループのオンラインメッセージを受信する場合、Nのマルチキャストグループに対応するオンライン用ポート番号を受信用として定義してください。

なお、上記Nのマルチキャストグループのオンライン用ポート番号は、データフィールドで同一番号としてください。

NXACPでは、マルチキャストグループ1つに対し、オンライン用/テスト用と2つポート番号を対応させます。

メッセージ入出力制御の組み合わせは、`nx_dfup()`のパラメータで指定しますが、この場合、各マルチキャストグループに対応するオンライン/テストモード用のポート番号定義との対応を、データフィールドの全ノードで統一させ、このモードとマルチキャストグループ番号とポート番号の組み合わせがデータフィールドで統一されていることで必要メッセージの受信、不要メッセージのOSレベルでの廃棄が可能となります。

NXACPが実際に使用するポートは、`nx_dfup()`のパラメータでユーザが指定したモードに対応するポートだけです。

表 2-13 モードとポート番号

MGN	オンラインモードポート番号	テストモードポート番号
1	55001	57001
23	55023	57023
99	55099	57099
101	55101	57101

NXACPが送受信時に使用するポート番号は、`nx_dfup()`のパラメータで指定したモードに対応するポート番号。

この対応をデータフィールド内の全ノードで統一する。

4.2 テスト形態

メッセージの入出力制御を利用したテスト形態の例を以下に示します。

(1) オンラインメッセージ受信によるテスト

S10VEのユーザプログラムを増設/変更する場合、テスト専用の新たなシステムを設定することなく、既存のオンラインメッセージのまま流用してテストできます。

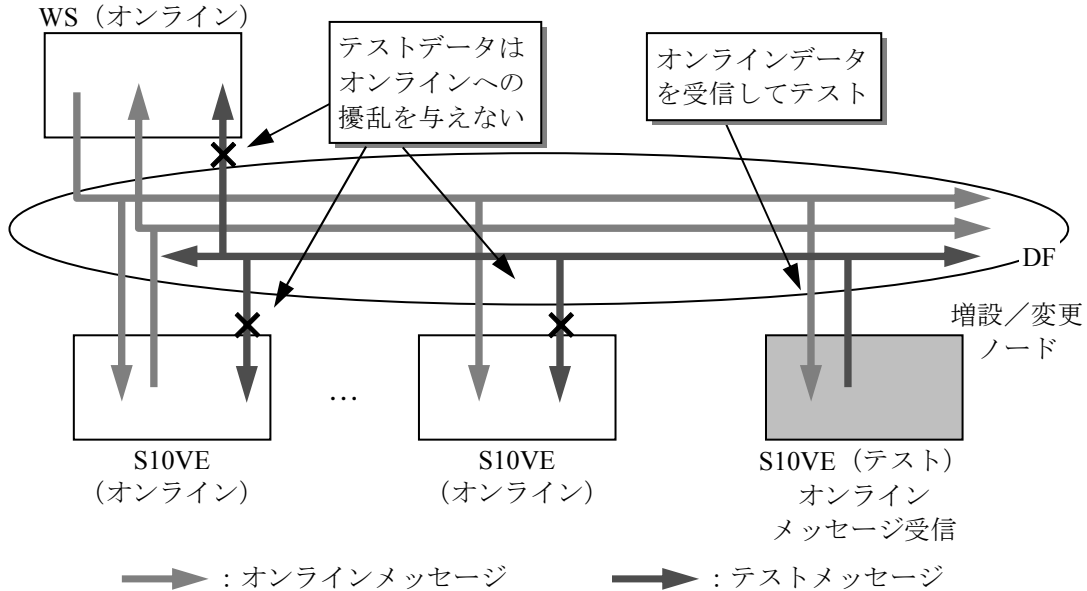


図2-38 テスト形態 (1)

(2) テストメッセージ受信によるテスト

S10VEとWSを新規追加する場合などで、WSからテストメッセージを受信し、その返答をWSに返すような場合でも、新たなテスト環境を設定する必要がなく、オンラインネットワークに直接テスト用WSとS10VEを接続してテストできます。

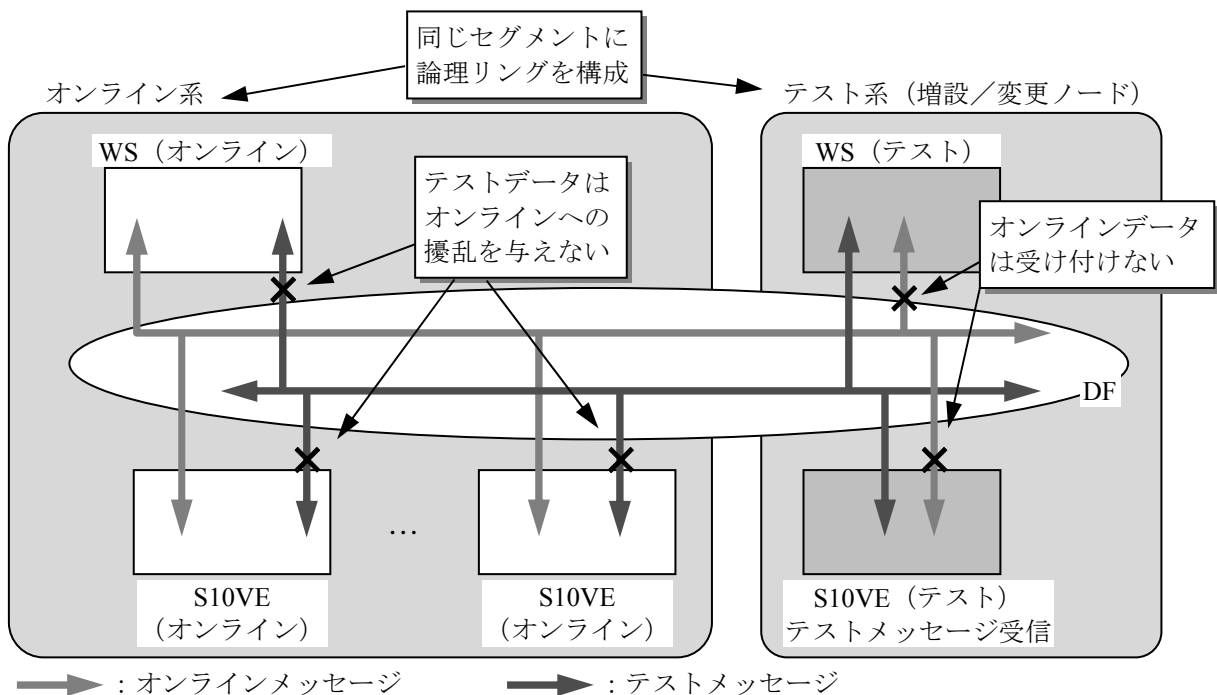


図2-39 テスト形態 (2)

(3) テスト/オンラインメッセージ受信によるテスト

S10VEとWSを新規追加する場合で、オンラインWSからオンラインメッセージを受信し、その返答メッセージをテストWSに返すテストができます。NX Dlinkのジャーナル機能を使用すれば、オンライン系のS10VE処理結果とテスト系のS10VE処理結果の照合も容易に行えます。

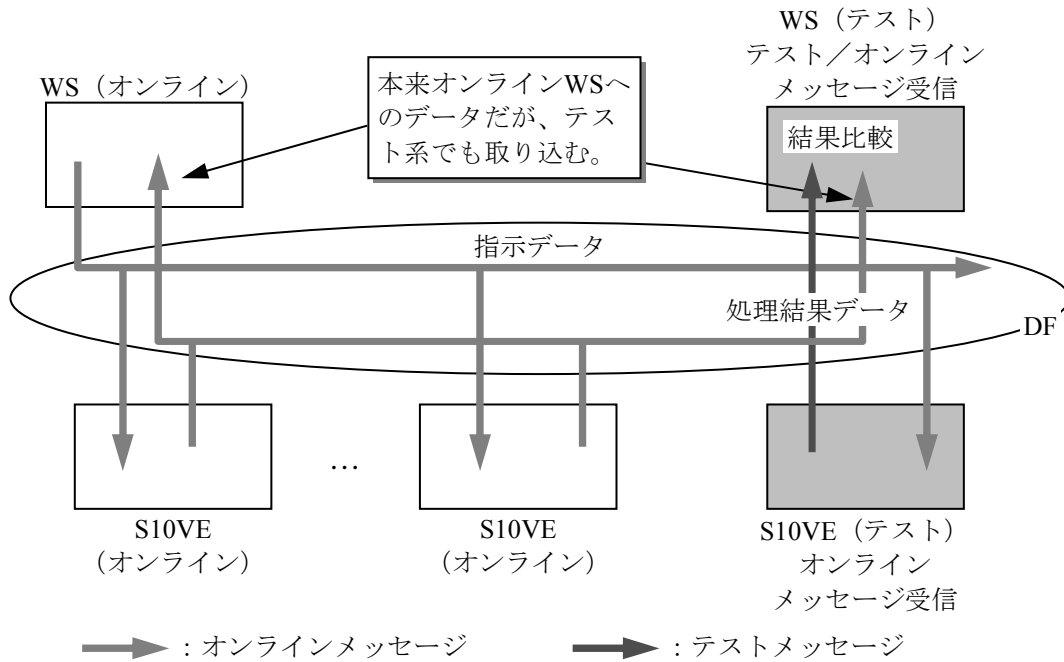


図2-40 テスト形態 (3)

4. 3 リモートデータフィールドとモード

ノードモードはデータフィールド単位に設定できます。したがって、データフィールドに送信される送信メッセージモードもデータフィールド単位に設定できます。

例えば、下図のようにローカルデータフィールド（DF1）でのノードモードをオンラインに設定し、リモートデータフィールド（DF2）でのノードモードをテストに設定した場合、DF1にブロードキャストされるメッセージモードはオンラインモードになり、DF2にブロードキャストされるメッセージモードはテストモードになります。

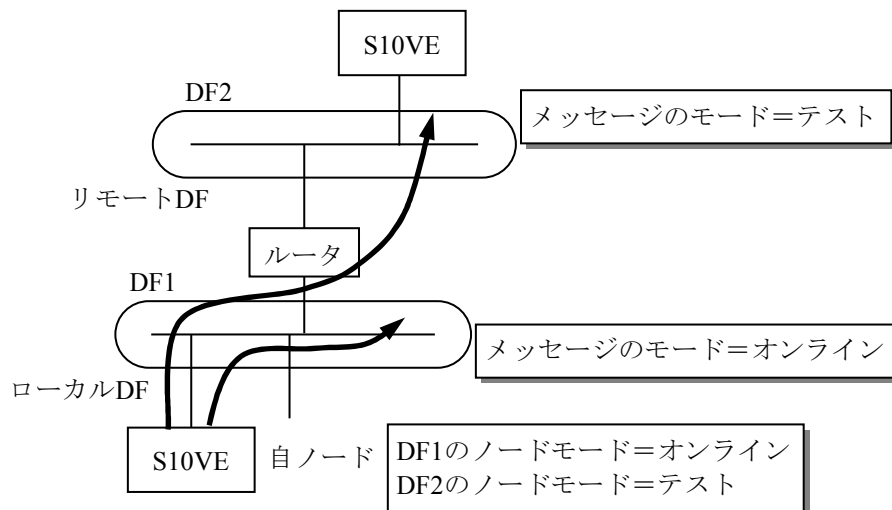


図 2-41 リモートデータフィールドとモード (1)

リモートデータフィールドからのメッセージを受信するノードでは、リモートデータフィールドの送信元ノードのノードモード（すなわち送信されるメッセージのモード）と、自ノードが受信するメッセージのモードを一致させてください。

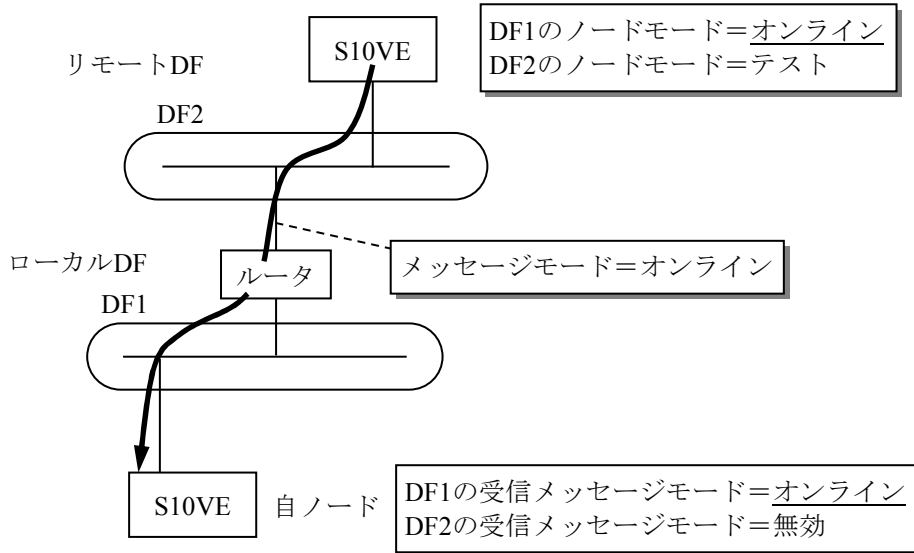


図2-42 リモートデータフィールドとモード (2)

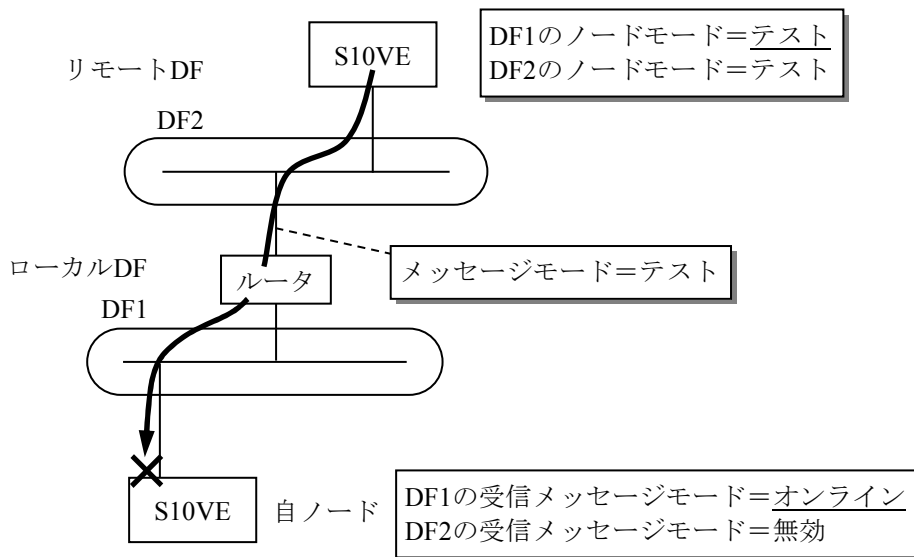


図2-43 リモートデータフィールドとモード (3)

第5章 システム管理機能

NXACPでは、障害やノードの状態変化を検知した場合、障害の内容と詳細情報をEASに知らせます。これを障害通知機能と呼びます。また、ユーザタスク管理機能として、メッセージを受信するユーザタスクの状態を監視し、メッセージを受信できない状態の場合は、NXACPでメッセージを廃棄します。

これらの機能を纏めてシステム管理機能と呼びます。

また、上記以外にも、DHPコードのトレースや制御トレース機能があります。

DHPトレースについては「付録D DHPレコード一覧」を、制御トレースについては「付録E 制御トレースについて」を参照してください。

5.1 障害通知機能

NXACPでは、以下に示すイベント発生時、EASまたは専用IRSUBにイベントの詳細情報を渡します。

EASの詳細については、「S10VE ソフトウェアマニュアル CPMS概説&マクロ仕様（マニュアル番号 SEJ-3-201）」を参照してください。

表 2-14 EAS通知イベント一覧

イベント名	イベント内容
ノード状態変化	ノードの状態に変化（生→死または死→生）がありました。
バッファステータス報告	送受信バッファの使用率がしきい値を超過したか、またはオーバフローしました。
プロトコル異常	受信したメッセージのNXヘッダに異常を検知しました。
ソケット障害	ソケット関連の障害を検知しました。
転写エリア重複異常	転写メモリ内の同じエリアに対し、2つ以上のノードで書き込み定義を検知しました（ソフト転写だけ）。

各イベントのデータ詳細は「付録B ログフォーマット」、「付録C ノード状態変化通知フォーマット」を参照してください。

【補足】

内蔵Etherのネットワークモジュールの障害については、RCTLNETがEASに通知します。

ネットワークモジュールの障害およびその対応方法は、「ユーザーズマニュアル S10VE総合編（マニュアル番号 SEJ-1-001）」を参照してください。

(1) ノード状態変化

ノード状態変化とは、他ノードからの生存信号を受信する場合にだけ（構築時指定）ユーザに通知するもので、他ノードの状態が生から死に変化した場合や死から生に変化した場合に報告します。二重化LANの場合は、変化を検知したLAN番号もユーザに通知しますので、ユーザは、どちらのLANに障害が発生したのかも、この通知で検出できます。

この通知は、変化した状態によって、以下のとおり検出タイミングなどが異なります。

● 死から生に変化した場合

[条件]

ノード立ち上げ時から最初に生存信号を受信した場合か、一度、死と判断した後、再度生存信号を受信した場合に通知します。

[検知タイミング]

ネットワークから生存信号受信時のノード状態変化チェック処理で検知します。

● 生から死に変化した場合

[条件]

通常モードの生存信号を受信した後、SHUTDOWN予告またはメンテナンス予告の生存信号を受信した場合か、通常モードの生存信号を受信した後、生存信号タイムアウト監視でタイムアウトとなった場合に通知します。

[検知タイミング]

SHUTDOWN予告またはメンテナンス予告の生存信号については、ネットワークから生存信号受信時のノード状態変化チェック処理で検知します。タイムアウトについては、1秒周期で動作するノード状態変化監視タスクのノード状態変化チェック処理で検知します。

【留意事項】

NXACPでの他ノード状態監視は、オプション機能で、ユーザの指定（構築情報）となります。これは、コントローラでの生存信号受信およびノードの状態変化監視の負荷を低減できるようにするためです。また、状態変化通知は専用IRSUB（#332）に報告します。他ノードの状態変化を監視する必要のないノードでは、ノード接続状態の監視機能をOFFで指定し、動作させることを推奨します。

WSやPCが接続するネットワークシステムでは、ノードの状態変化監視は、できるだけWSまたはPCで行うことを推奨します。WSやPCが接続しないネットワークシステムでノード状態変化を監視する必要がある場合は、特定コントローラでノードの状態変化を監視し、他のコントローラでは、生存信号を受信しない運用を推奨します。

また、ノードの状態変化を監視するコントローラは監視専用コントローラとし、将来追加される機器の負荷まで含めたシステム設計を推奨します。

二重化LANシステムでは、この変化通知によって、ノード単体の障害なのかネットワークの障害なのか切り分け、システム全体の監視をすることが可能となります。システム監視の例を表2-15に示します。

例として、図2-44のシステム構成での障害監視パターンを示します。

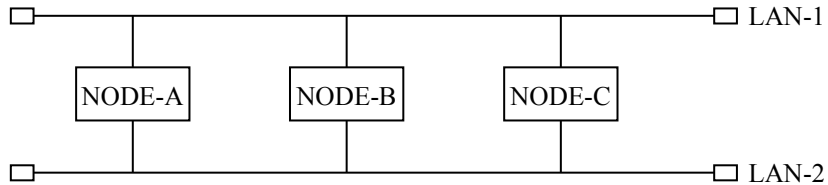


図2-44 構成例

表2-15 障害監視例

障害情報 (NXACPが通知する情報)	障害部位	
	障害図	障害図
<ul style="list-style-type: none"> • NODE-A (LAN1) の状態が死に変化。 • NODE-A (LAN2) の状態は生のまま (変化通知なし)。 • NODE-A以外のノード (LAN1) は生のまま (変化通知なし)。 		<ul style="list-style-type: none"> • NODE-AのLAN-1側の内蔵Ether • LAN伝送路 (ブリッジ、リピータ含む)
<ul style="list-style-type: none"> • NODE-Aの状態がLAN1、LAN2共死に変化。 • 他ノード (Node-B) 状態は生のまま (変化通知なし)。 		<ul style="list-style-type: none"> • NODE-A
<ul style="list-style-type: none"> • NODE-A、B (LAN1) の状態が死に変化。LAN1側の全ノードが死に変化。 • NODE-A、B (LAN2) の状態は生のまま (変化通知なし)。 		<ul style="list-style-type: none"> • NODE-CのLAN-1側の内蔵Ether • LAN伝送路 (ブリッジ、リピータ含む) ※LAN伝送路か内蔵Etherかは、内蔵Etherの障害通知で切り分け可能。

障害情報は、NODE-Cに対する通知情報を示します。

(2) バッファステータス報告

バッファとは、メッセージの送受信に使用するメッセージ格納メモリのことで、データフィールドごとに送信用と受信用に分けて管理しています。このようにバッファを分離して管理することで不当なユーザタスクによるバッファ占有範囲を限定し、他業務への異常波及を防止しています。

バッファ面数は、構築時にユーザが指定します。

NXACPでは、上記バッファがオーバーフローした場合、またはしきい値を超過（低下含む）した場合、イベントをログし、1秒周期でログデータをユーザに報告します。以下にそのタイミングを示します。

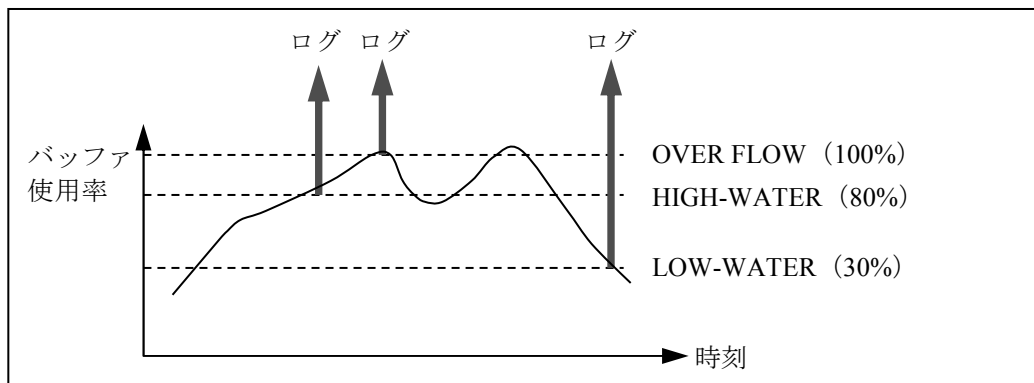


図2-45 バッファステータス出力タイミング

[条件]

オーバーフロー：LOW-WATER超過後、最初にバッファ確保不可となったときにログします。

HIGH-WATER：LOW-WATER超過後、最初にHIGH-WATER超過時ログします。

LOW-WATER：HIGH-WATERまたはオーバーフロー発生後、最初にLOW-WATERを下回ったときにログします。

[検知タイミング]

送信バッファは、ユーザからのメッセージ送信要求に伴うバッファ確保時に検知します。

受信バッファは、ネットワークからメッセージ到達に伴う受信バッファ確保時に検知します。

[条件を満たしたときの処理]

HIGH-WATER、LOW-WATERについては、条件を満たした場合でも、NXACPの動作に変化はありません。ただし、オーバーフロー発生中は以下の動作をします。

送信バッファ：ユーザタスクからの送信要求は、すべてバッファ確保不可としてエラーリターンします。

受信バッファ：ネットワークからのメッセージ受信処理を空読みします。つまり、受信バッファオーバーフロー中は、到着メッセージを廃棄します。

[ユーザ対処]

通信量を低減、またはバッファ容量を見直してバッファを拡張してください。

【補足】 HIGH-WATER、LOW-WATERのしきい値は構築情報で変更できます。

(3) プロトコル異常

プロトコル異常とは、送信する際に各メッセージに付加されるNXヘッダ異常のことです。これは、送信／受信ノードでの構築誤りなどによって発生する可能性があります。

NXACPでは、上記障害を検知した場合、検知したヘッダ情報をそのままユーザに報告します。

[条件]

ヘッダ情報に異常を検知した場合に通知します。

[検知タイミング]

ネットワークからメッセージ受信時、ヘッダチェック処理で検知します。

[条件を満たしたときの処理]

異常を検出したメッセージは、EAS報告後NXACPで廃棄します。

[ユーザ対処]

送信元ノードとの構築情報の見直しを実施してください。

(4) ソケット障害

RCTLNETでのEASへの障害報告はハードウェア障害だけで、ソケットインタフェースに絡む異常はソケットマクロ発行元にその詳細情報を返します。NXACPでは、この詳細情報をEASに通知します。

このエラーは、現状ではリモートデータフィールドのネットワークアドレス定義誤りによるものだけです。

[条件]

ソケットマクロ (`sendto()`のENETUNREACH) がエラーリターンした場合に通知します。

[検知タイミング]

オンライン処理中のソケットマクロ (`sendto()`) 発行後のリターンコードで検知します。

[条件を満たしたときの処理]

異常になった送信要求は無効となります。

[ユーザ対処]

構築情報 (リモートデータフィールドのネットワークアドレス) を見直してください。

(5) 転写エリア重複異常

複数のノードが、転写メモリ内の同じアドレスに対し、書き込みエリアを定義した場合に通知します。重複チェックを実施するかどうかは、イニシャライズ時にユーザが指定してください。

[条件]

転写メモリ内の同じアドレスに対し、複数のノードが書き込みをした場合に通知します。

[検知タイミング]

他ノードが送信したソフト転写データを受信したときに検知します。ただし、チェックを行うのは、自ノードのソフト転写の状態が、送受信開始か受信処理開始のどちらか以降の場合だけで、開始前に通知することはできませんので注意してください。

[条件を満たしたときの処理]

重複を検知したソフト転写受信データは無効となり、転写メモリに反映されません。

[ユーザ対処]

ソフト転写メモリのエリア分割を見直してください。

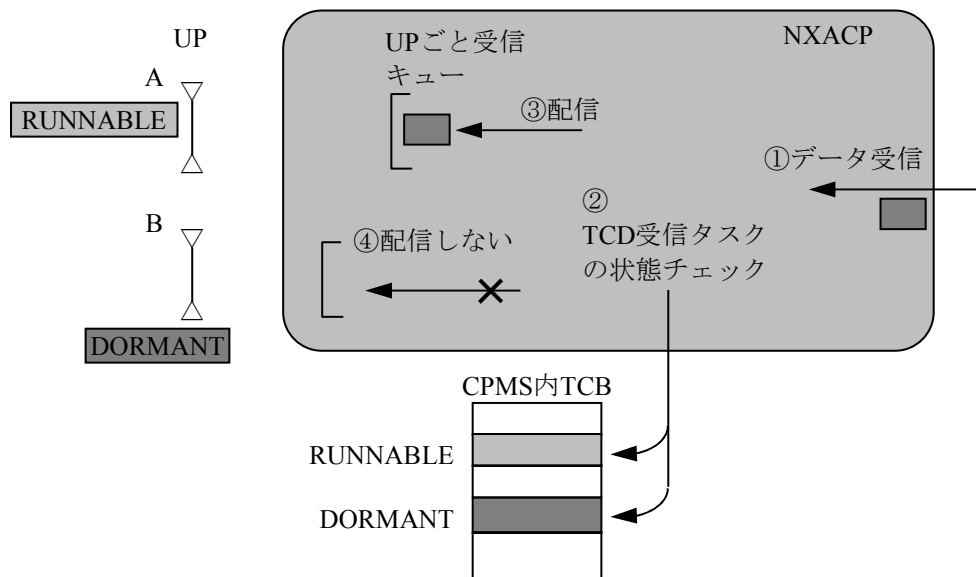
5.2 ユーザタスク管理機能

受信バッファは、ネットワークからデータ受信時に確保し、ユーザタスクがnx_get()を発行し受信したタイミングで解放します。したがって、ユーザタスクがnx_get()を発行しない場合、受信バッファはオーバフローしてしまいます。

NXACPでは、受信バッファのオーバフローを防止するため、ユーザタスクの状態を監視し、ユーザタスクがnx_get()を発行できない状態の場合は、該当タスクの受信待ちメッセージや到着メッセージを廃棄します。

(1) メッセージ到着時のチェック

メッセージ到着時、該当トランザクションコードを入力するユーザタスクの状態をチェックし、DORMANT/IDLE/NON-EXIST状態の場合、メッセージ配信をしません。また、入力する予定のユーザタスクがすべて上記状態の場合は、到達したメッセージを廃棄します。



RUNNABLE状態 (nx_get()でメッセージ待ち中または受信処理中) のタスクAにはメッセージを配信しますが、DORMANT状態のタスクBには配信しません。

図2-46 メッセージ到着時のタスク状態チェック

(2) DORMANT遷移時のチェック

ユーザタスクがDORMANTに遷移時、該当ユーザタスクの未受信メッセージをチェックし、未受信メッセージがあれば受信キューからページします。また、入力する予定のユーザタスクがすべて受信完了している場合、またはすべてDORMANTになった場合は、メッセージを廃棄します。

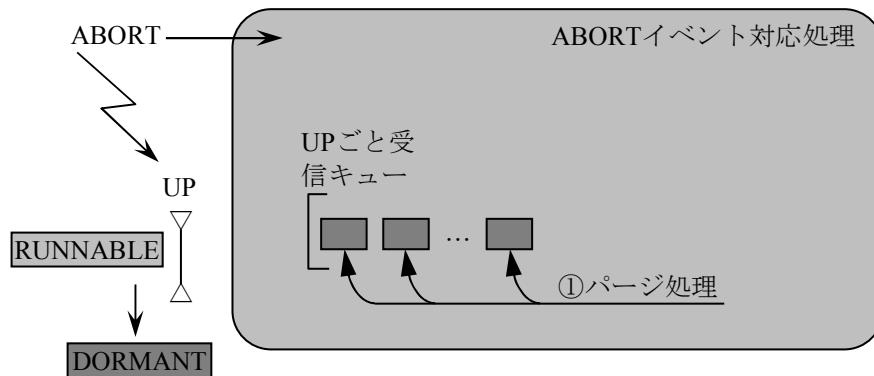


図 2-47 ABORT後処理

ユーザタスクがIDLEに遷移したタイミングでの上記受信キューのページは行いません。これは、以下によるものです。

- 通常nx_get()を発行するユーザタスクは、デーモン型のプログラム構造となりますので、IDLE状態にはならないこと
- EXITのシステム組み込みサブルーチンに処理を組み込んだ場合、負荷が増大すること
- 次メッセージが到着すれば、「(1)メッセージ到着時のチェック」タイミングでページ処理が動作すること

第6章 運用管理機能

NXACPの運用を以下に示します。この章では、「NXACPの立ち上げ」以降の運用について説明します。

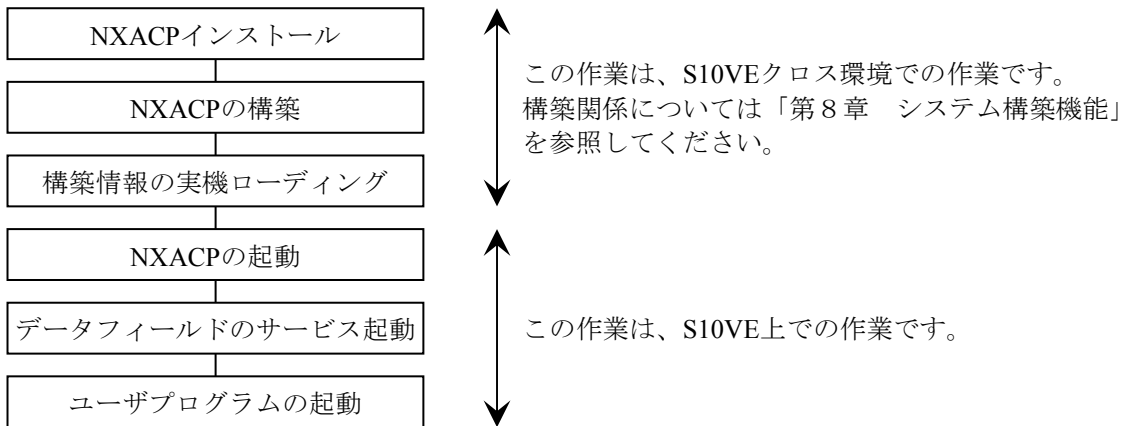


図2-48 NXACPの運用

S10VE実機でのNXACPの運用には以下があります。

- ・NXACPの起動/停止
- ・データフィールド単位のサービス開始/停止
- ・ネットワークモジュールの抜去/挿入

NXACPの起動では、データフィールドのサービス開始に先立って、NXACP（プログラム本体）を起動する必要があります。NXACP（プログラム本体）の起動は、システム立ち上げ時、初期化処理およびデータフィールドの立ち上げ処理を、ユーザプログラムの起動前に一度だけ行ってください。

データフィールドのサービス開始とは、データフィールド単位の送受信処理の開始を指します。構築時に定義した各データフィールドは、このサービス開始指示をもって送受信処理の受け付けを開始しますので、定義したデータフィールド各々に対してサービス開始要求を発行してください。

NXACP、データフィールドのサービスは、必要に応じて処理を停止することもできます。

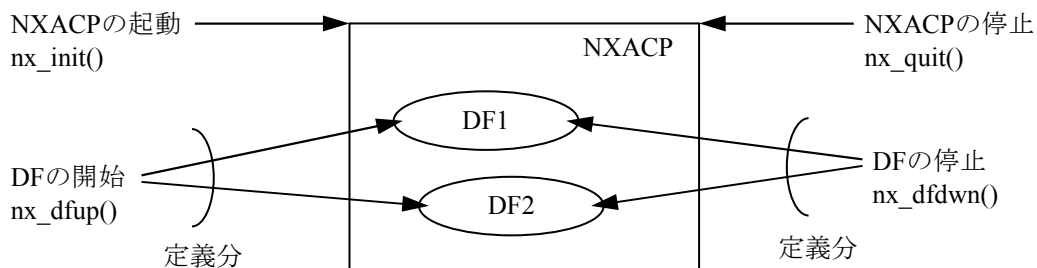


図2-49 NXACPとデータフィールドの運用

6. 1 NXACPとデータフィールドの運用管理

NXACPの起動/停止は、それぞれnx_init()、nx_quit()マクロを用いて行います。

nx_init()マクロ発行によって、データフィールドの運用を開始できるようになります。ただし、nx_init()マクロを発行しただけでは、データフィールドでの送受信は行えません。nx_init()マクロ発行後、各データフィールドに対し、nx_dfup()マクロを起動してください。

nx_quit()マクロ発行によって、すべてのNXACPのサービスが停止します。起動中のデータフィールドもすべて停止され、データフィールドとのメッセージ送受信が停止します。各データフィールドを個々に停止したい場合には、nx_dfdown()マクロを使用します。

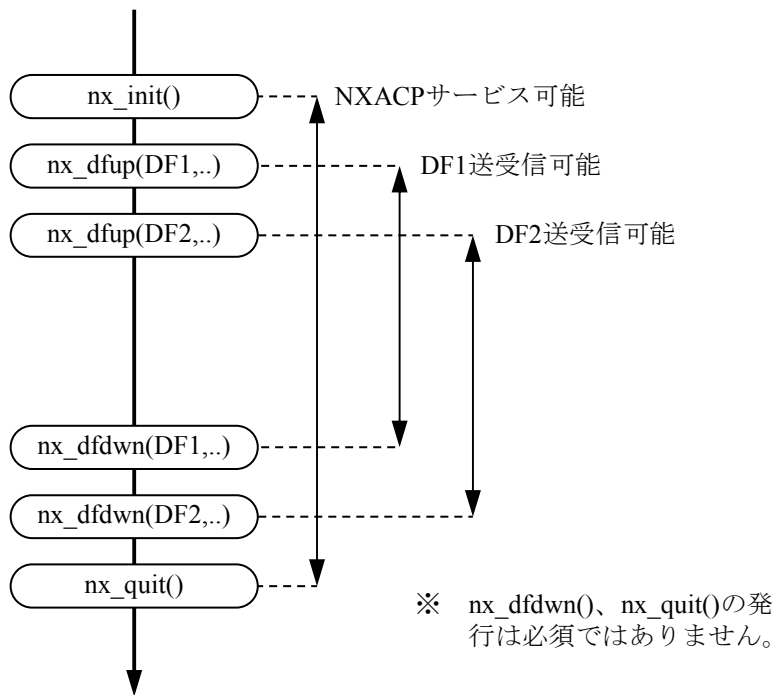


図 2-50 各マクロと運用

6. 1. 1 データフィールド起動

ユーザタスクがデータフィールドを介してメッセージを送受信するには、`nx_dfup()`を用いてそのデータフィールドを起動する必要があります。

NXACPではデータフィールドの種別として、以下の3種類があります。

- ・自ノード内データフィールド
- ・ローカルデータフィールド
- ・リモートデータフィールド

データフィールドの種別にかかわらずデータフィールドの起動には`nx_dfup()`を用います。

`nx_dfup()`を発行すると、対象データフィールドは以下のような処理をします。

表 2-16 データフィールド起動処理

	自ノード内DF	ローカルDF	リモートDF
対象DFへのメッセージ送信準備	○	○	○ (*)
対象DFからのメッセージ受信開始	○	○	○ (*)
対象DFへの生存信号送出開始	—	○	—

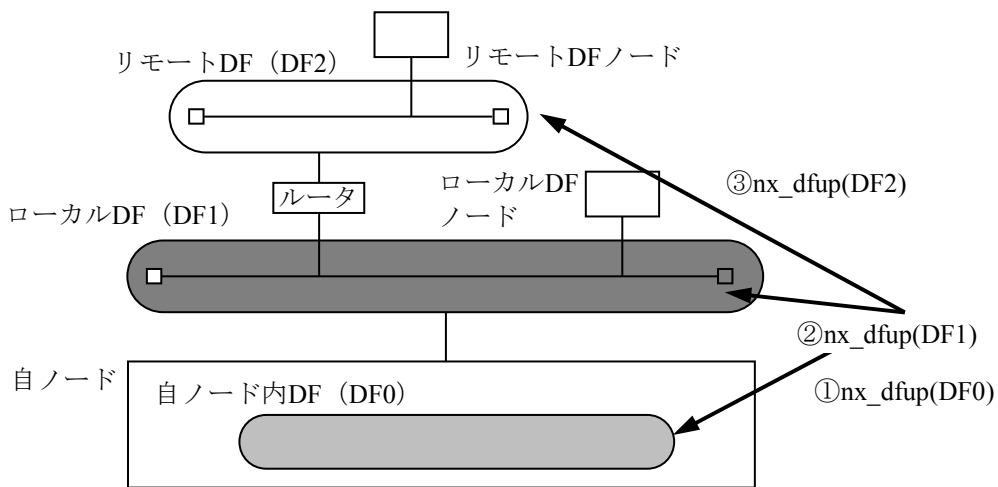
○：処理する —：処理しない

(*) リモートデータフィールドへの送信、リモートデータフィールドからの受信は、ローカルデータフィールドが起動されていて初めて動作します。

自ノード内のユーザタスク間でメッセージ通信をしたい場合には、自ノード内データフィールドを用いるため、自ノード内データフィールド (DF#=0) を起動する必要があります。

リモートデータフィールドを使用する場合には、使用するリモートデータフィールドに接続されるローカルデータフィールド (ソースデータフィールドと呼びます) を起動してから、その後リモートデータフィールドを起動してください。なお、ローカルデータフィールドを起動しない状態のまま、リモートデータフィールドからのメッセージはNXACPレベルで廃棄します。

また、送信要求に対しても、エラーと判定し処理を中断します。



※ ローカルDF、リモートDFの順に起動してください。
(`nx_dfup(DF1)`、`nx_dfup(DF2)`の順)

図 2-51 データフィールドの初期化順序

【留意事項】 起動失敗時の扱いについて

データフィールドを1LANで構成している場合、ポートの生成/設定エラーなどの異常を検知すると、データフィールドの起動失敗としてエラーリターンします。異常を検知すると、そのデータフィールドの起動処理を中止し、`nx_dfup()`発行前の状態までネットワーク関連の設定を戻し（途中まで生成したポートは消去します）、データフィールドを停止状態に戻した後、制御をユーザに戻します。

このとき、EASへの報告、エラーログは行いません。内蔵Etherでの障害発生については、RCTLNETでEASに障害通知しますので、どの部位で障害が発生したかについては、`nx_dfup()`のリターンコードおよびエラーログ情報から分析できます。

また、データフィールドを二重化LANで構成している場合、両方のLANの設定でエラーが発生したときにも、そのデータフィールドの起動処理を中止し、`nx_dfup()`発行前の状態までネットワーク関連の設定を戻し（途中まで生成したポートは消去します）、データフィールドを停止状態に戻した後、制御をユーザに戻します。

`nx_dfup()`がエラーリターンしても、他の起動中のデータフィールドには影響はありません。

6. 1. 2 データフィールド停止

オンライン中にメッセージ送受信を停止したい場合には、データフィールドを停止させてください。

また、ユーザプログラムを動作させたまま、ノードのモードまたは受信するメッセージのモードを変更する場合も、データフィールドを一時停止してから再起動してください。

データフィールドの種別にかかわらず個々のデータフィールドの停止には`nx_dfdwn()`を用います。

なお、`nx_quit()`を用いてNXACPを停止することによって、すべてのデータフィールドを一度に停止することもできます。構築データを変更する場合は、構築情報をローディングするコマンドで、NXACPを停止させますが、この場合、SHUTDOWN予告の生存信号を送信しません。他ノードに計画停止であることを通知する場合、`nx_quit()`を用いてNXACPを停止させてください。

`nx_dfdwn()`を発行すると、以下の停止処理をします。

表 2-17 データフィールド停止処理

	自ノード内DF	ローカルDF	リモートDF
対象DFへのメッセージ送信停止	○	○	○
対象DFからのメッセージ受信停止	○	○	○
対象DFへの生存信号送出停止	—	○ (*)	—

(*) 生存信号をshutdown予告モードで複数回送信した後、生存信号の送出を停止します。

停止するデータフィールドの順序には、特に制限はありません。

ただし、ローカルデータフィールドを停止すると、そのデータフィールドに接続されたリモートデータフィールドも自動的に停止します。このときユーザは、このリモートデータフィールドへの`nx_dfdwn()`発行は必要ありません。

なお、このローカルデータフィールドを再び起動しても、このリモートデータフィールドは自動的に起動されたりしませんので、再びリモートデータフィールドを起動する場合には、このリモートデータフィールドに対して`nx_dfup()`を発行してください。

`nx_dfdwn()`が発行された時点で処理待ちの各メッセージは廃棄されます。他の停止要求を発行しないデータフィールドでの送受信メッセージはそのまま正常に処理されます。ただし、`nx_dfdwn()`が発行された時点で、送信メッセージとしてRCTLNETに渡されたメッセージに関しては、廃棄せずそのままネットワークに送出されます。

`nx_dfdwn()`で指定されたデータフィールドがローカルデータフィールドの場合には、shutdown予告メッセージの送出処理をしますので、この送出処理に約10秒かかります。したがって、複数のデータフィールドを連続して停止するために`nx_dfdwn()`を用いる場合は、2つ目以降のデータフィールドの停止処理が開始するまでに約10秒かかりますので注意してください。

なお、同時に複数のデータフィールドを停止したい場合には、`nx_quit()`を用いることができます。`nx_quit()`を用いた場合でも、起動されていたローカルデータフィールドには、shutdown予告メッセージの送出処理をします。なお、`nx_quit()`を用いるとNXACPも併せて停止しますので、再度NXACPを用いたい場合には、`nx_init()`を再発行してください。

【留意事項】

- ・停止失敗時の扱いについて
`nx_quit()`、`nx_dfup()`は、パラメータチェック以外の異常を検知してもエラーリターンしません。
 データフィールドを停止状態にし、制御をユーザに戻します。`nx_quit()`、`nx_dfdwn()`でのエラー発生は、パラメータ異常や起動中でないデータフィールドへの停止要求以外では、内蔵Ether障害発生が原因と考えられます。内蔵Etherでの障害発生については、RCTLNETがEASで障害通知しますので、どの部位で障害が発生したかが分かります。また、障害の内容については、エラーログ情報から分析することができます。
- ・ローカルデータフィールド停止に伴うリモートデータフィールドへの影響について
 ローカルデータフィールドを停止した場合には、そのローカルデータフィールドと接続されたリモートデータフィールドとのメッセージ送受信も停止します。

ローカルDFを停止すると、リモートDFへの送信も停止

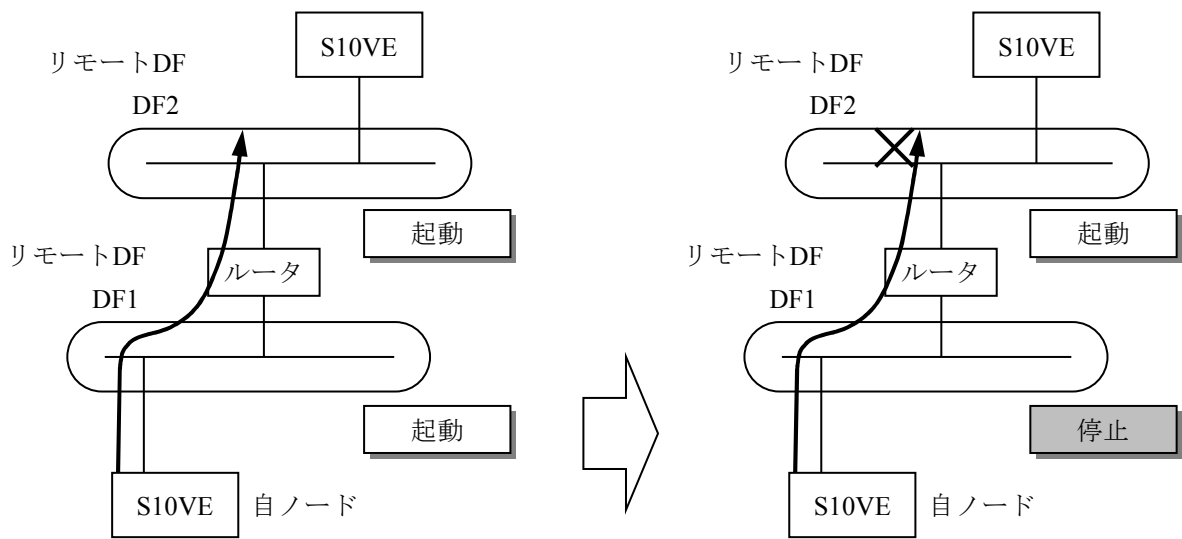


図2-52 リモートデータフィールドへの影響 (1)

ローカルDFを停止すると、リモートDFからの受信も停止

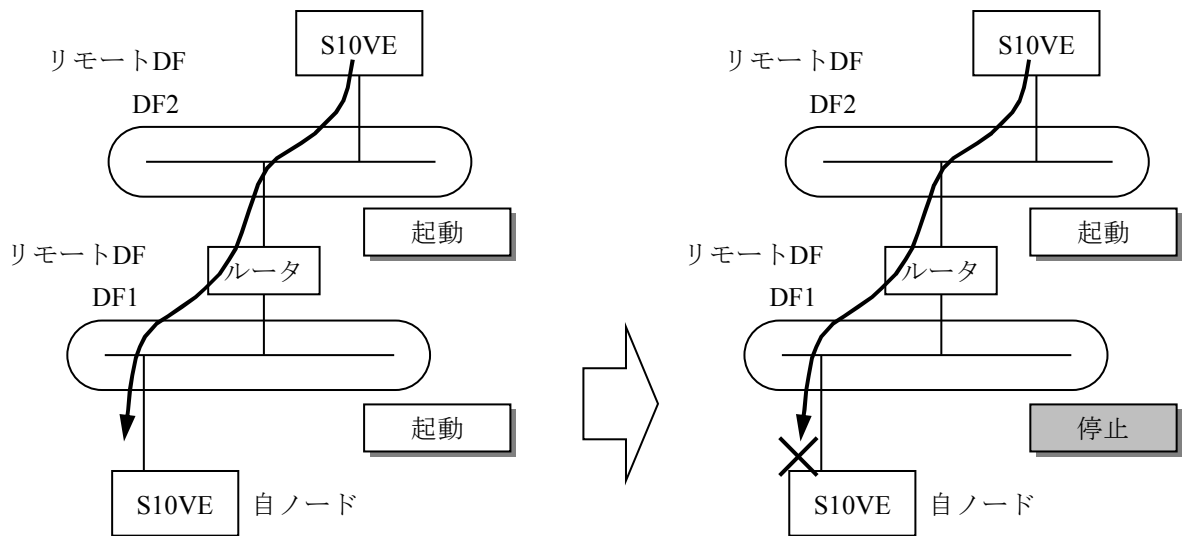


図2-53 リモートデータフィールドへの影響 (2)

6. 1. 3 モードの設定と変更

データフィールドのノードモード、および受信メッセージのモードは、データフィールドごとに設定できます。どのモードを使用するかは、`nx_dfup()`発行時に、そのパラメータとして指定します。このとき指定したモードは、そのデータフィールドが動作中は有効ですが、動作中に変更することはできません。

変更する場合には、一度該データフィールドを`nx_dfdwn()`を用いて停止させてから、`nx_dfup()`を用いて新たなモードを指定して再度データフィールドを起動してください。

モード設定には、ノードモードと、受信するメッセージのモードの2つの設定が必要ですが、モード変更がどちらか一方だけであっても、一度データフィールドを停止させてください。

このように、モード変更を`nx_dfdwn()`と`nx_dfup()`の2つのマクロで行うのは、モード変更に伴うユーザタスク側の種々の設定を容易に行えるようにするためです。モード変更に伴うテーブルおよびユーザタスクの初期化などは、`nx_dfdwn()`発行と`nx_dfup()`発行の間に行えます。

このときネットワークからの該データフィールド宛のメッセージの受信は停止していますので、バッファオーバーフローを気にすることなくモード変更準備を行えます。また、モード変更前のサービス待ち送受信メッセージはすべて廃棄され、`nx_dfup()`を発行するまでの間の到着メッセージも廃棄されますので、`nx_dfup()`発行前の状態を気にすることなく、新しいモードでデータフィールドとのメッセージ送受信を再開することができます。

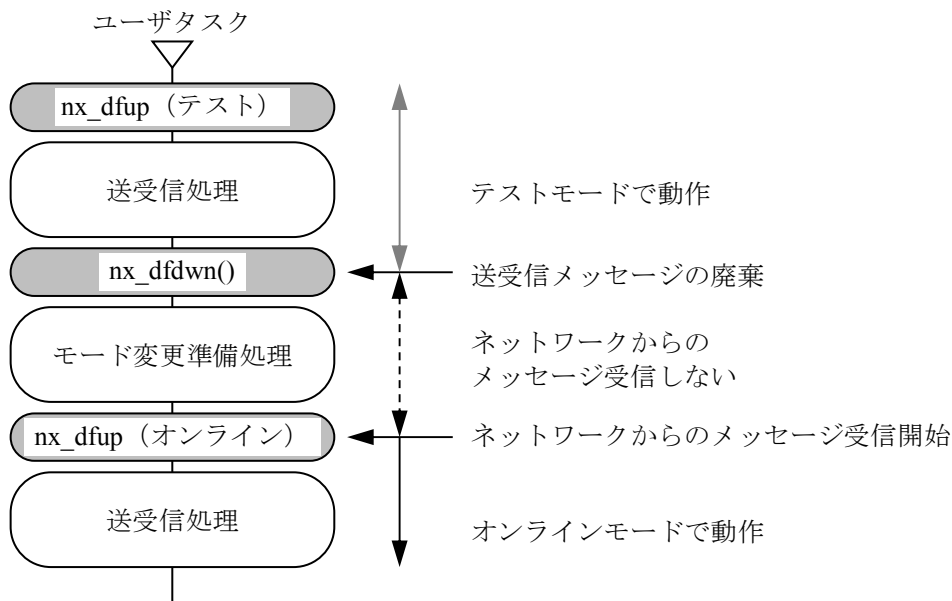


図 2-54 モード変更手順

【留意事項】

- ・ノードモードの設定は、自ノードデータフィールドでは無効です。
- ・受信するメッセージモード設定は、ローカルデータフィールドでだけ有効です。

表 2-18 モード指定の有効範囲

データフィールド種別	ノードモード設定	受信するメッセージモード設定
自ノード内データフィールド	×	×
ローカルデータフィールド	○	○
リモートデータフィールド	○	×

○：有効 ×：無効

6. 1. 4 構築情報の変更

NXACPの構築情報を変更する場合には、NXACPを停止します。

POCからの構築情報再ローディングは、`tblldnxsv`コマンドを用いてS10VEのサイトにローディングします。

ローディング時は自動的にNXACPを停止しますので、ローディング完了後は、`nx_init()`を用いてNXACPを再起動してください。

また、`tblldnxsv`コマンドではNXACPを停止させる際、強制終了させますので、SHUTDOWN予告の生存信号を送信しません。他ノードに対してSHUTDOWN予告が必要な場合、ユーザが再ローディング前に`nx_quit()`を発行してください。

第7章 ネットワーク共有メモリ機能

ネットワーク共有メモリ（以下、転写メモリと称します）とは、転写メモリに書き込んだデータを定周期で他ノードの読み出しエリアに転写する機能のことをいいます。同じアドレスに対し、ノード1台が書き込み可能で、複数台のノードで読み出しが可能です。書き込みノードは、静的にシステムで固定されます。

転写には、ハード転写とソフト転写の2種類があり、ユーザインタフェースはハード／ソフト転写でも共通のため、通信媒体を意識することなくメモリ転写機能を使用することができます。

ただし、ハード転写は、S10VEでは非サポートです。

7. 1 用語説明

以下に転写メモリで使用する用語について説明します。

(1) 転写メモリ

転写メモリとは、64バイトを1ブロックとした転写エリアで構成されます。

転写エリアのサイズは転写種別によって異なります。

(2) ソフト転写

転写メモリとしてグローバルエリアを使用し、実際の転写はNXACPのマルチキャスト通信を用いて行います。転写エリアのサイズは可変で、最大16384ブロックまで指定できます。

なお、最大ブロック数は、ユーザが転写メモリ初期化時に指定します。

転写の周期はミリ秒単位です。

ソフト転写には、周期時間に送信することを目的とした周期送信型と周期以内に一定の時間で分割送信する分割送信型の2種類あります。

周期送信型では、送信するメッセージに排他制御を行うことができますが、分割送信型では分割送信を行うため排他制御はできません。

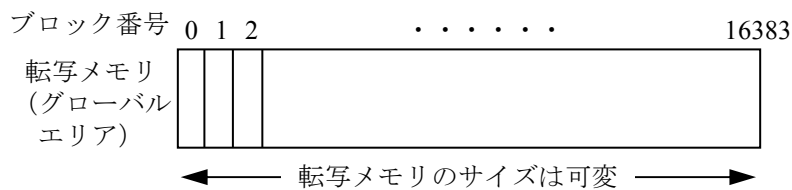


図2-55 ソフト転写メモリ

(3) 転写メモリ識別子 (Transfer Memory ID)

転写メモリは、データフィールドごとに複数定義できます。このときの各々の転写メモリエリアを認識するのが転写メモリ識別子 (TMID) です。

ソフト転写ではデータフィールドに4つ定義できます。

ユーザはTMIDを指定して、転写メモリにアクセスします。

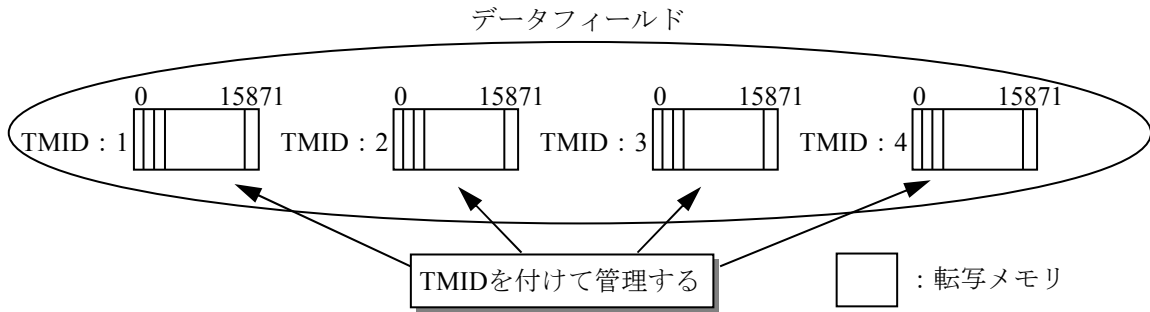


図2-56 転写メモリ識別子 (TMID)

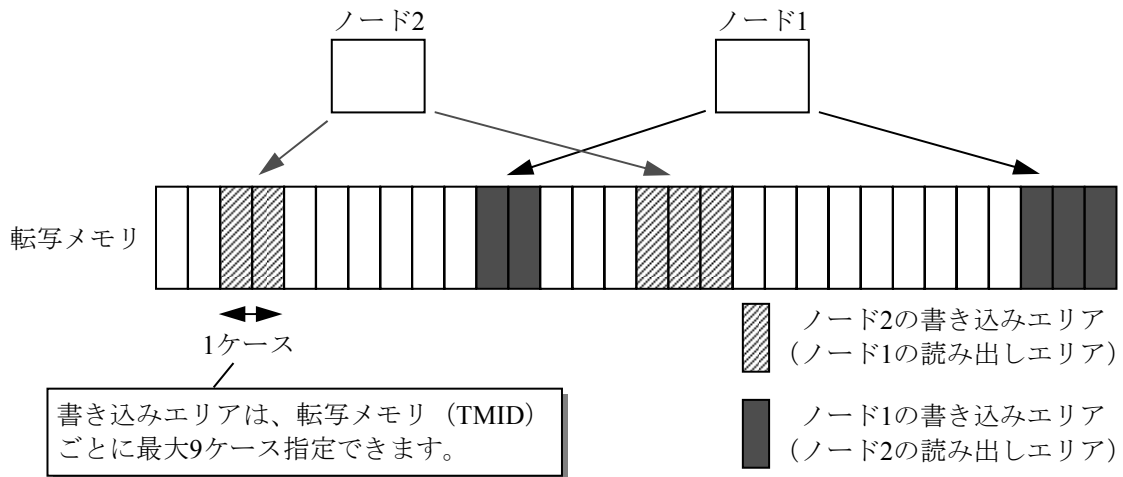
(4) 書き込みエリア/読み出しエリア

転写メモリエリアは、書き込みエリアと読み出しエリアに分けられます。

書き込みエリアは、自ノードが指定したデータをデータフィールドに送出するエリアのことで、転写メモリ初期化時に指定し、その数はTMIDごとに最大9ケースまで指定できます。

読み出しエリアは、他ノードのデータを受信するエリアのことで、読み出しエリアに関しては、特別な指定はありません。書き込みエリア以外のエリアは、読み出しエリアとなります。

なお、ケースとは連続したブロックを指します。



1ケースは、1~255ブロックの範囲で指定できます。

図2-57 書き込み/読み出しエリア

7. 2 転写メモリの仕様

(1) 転写メモリの識別と制御単位

転写メモリは、データフィールドごとに定義でき、転写メモリごとにユニークな転写メモリ識別子 (TMID) を持ちます。転写メモリの属性情報を以下に示します。

表 2-19 転写メモリの属性

属性	方式	ソフト転写
転写メモリ/データフィールド	4個	
サイズ/1転写メモリ	64バイト~1MB	
1ノードからの書き込みエリア数/ 転写メモリ	最大9個	
転写周期/書き込みエリア	(*3)	
書き込みエリアの重複チェック	データ受信時に検知。 EASに通知。	
メモリ転写の制御者	NXACP	
プロテクション	すべて読み書き可能 (*1)	
排他制御	16320バイト (*2)	
転写メモリのマッピングエリア	グローバルエリア CMエリア	
二重化LAN構成時の陰蔽	NXACP	

(*1) 読み出しエリアに書き込んだ場合、他ノードからのデータが周期的に書き込まれるため、無効となります。

(*2) 直接メモリにアクセスするときは4バイト保証です。
転写メモリの排他制御機能を使用した場合が16320バイトまで保証します。

(高速ソフト転写は、排他制御機能がありません。)

(*3) 周期送信型は100ミリ秒~86400秒、分割送信型は10ミリ秒~86400秒です。

(2) TMIDとMCGの関係

転写はNXACPのマルチキャスト通信を用いて行います。このため、NXACPの構築情報のマルチキャストグループを定義してください。また、負荷を下げるためTMIDごとに専用のマルチキャストグループを使用することを推奨します。

<ソフト転写使用例>

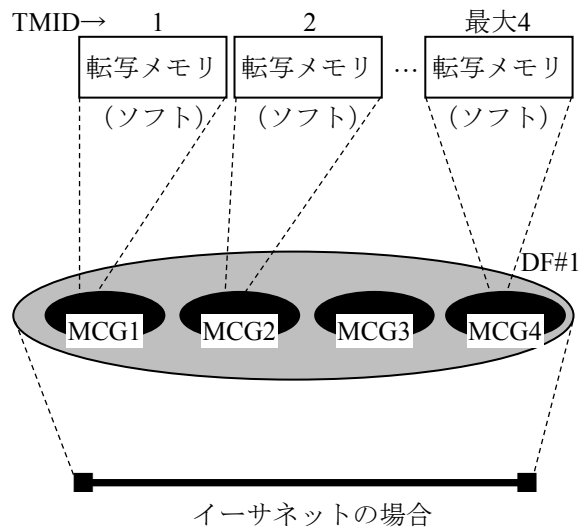


図 2-58 転写メモリ認識とマルチキャストグループ

(3) 排他制御

転写メモリの排他制御は、初期化要求時のオプションとしてユーザが指定します。

ここで述べる排他制御とは、ユーザ指定読み書き要求とNXACPの送受信処理間の転写メモリに対する排他制御のことで、この排他制御を使用することで16320バイトまで整合性を保証します。

なお、排他制御はTMIDごとに行います。

<排他制御を用いた転写メモリ>

<直接メモリアクセスした転写メモリ>

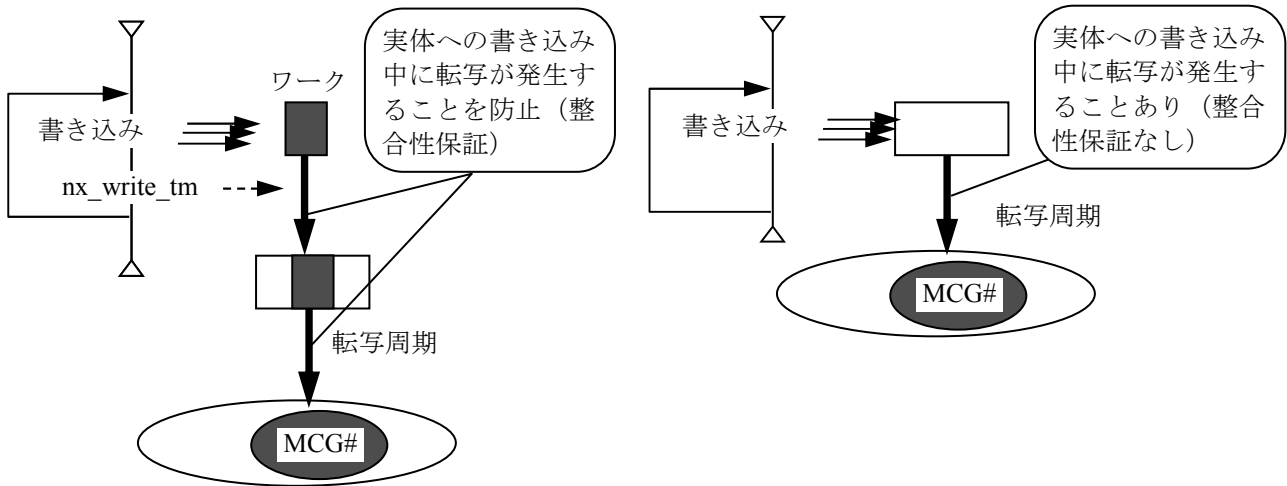


図 2-59 ソフト転写時の排他制御

【留意事項】

排他制御はオプション機能で、使用しない場合は4バイトまでの保証となります。

排他制御を使用しないケースでは、ユーザが書き込み中に、中途半端なデータのまま他ノードの転写メモリに書き込まれてしまいますので注意してください。

高速ソフト転写は、定周期を維持するため、排他制御はありません。

(4) 書き込みエリアの重複チェック

メモリ転写では、転写エリアに対し書き込みエリアを各ノードで重複しないように設定することによって、書き込み/読み出しのタイミングをユーザで取れるようになります。NXACPでは、各ノードの書き込みエリアの重複チェックを以下の方法で実施します。

なお、書き込みエリアの重複チェック有無は、ソフト転写では初期化要求時のオプションとしてユーザが指定します。

・ソフト転写

他ノードが送信したデータを受信したときにチェックし、重複検知時にEASで報告します。重複検知時の受信データは廃棄します。重複を検知した後も、重複した各ノードでの書き込み要求はエラーリターンしませんので注意してください。

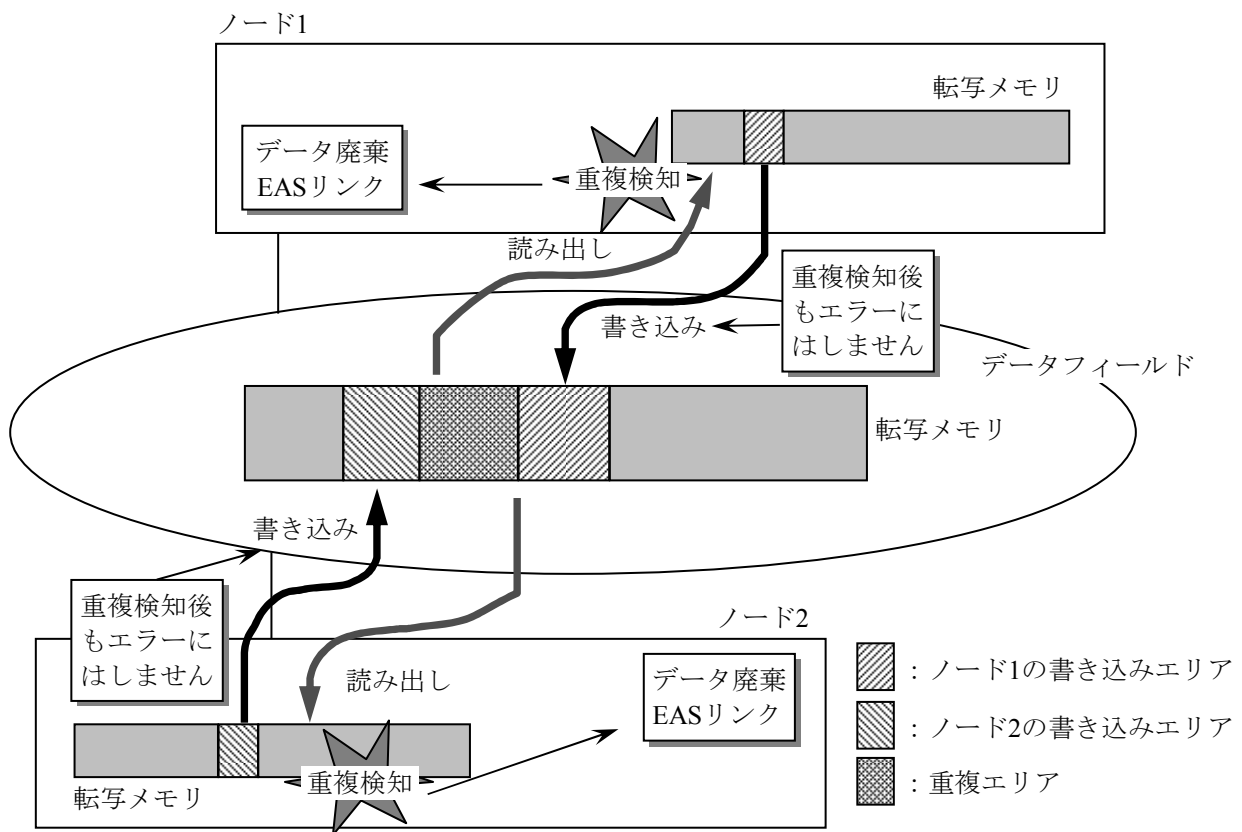


図 2-60 ソフト転写時の重複チェック

自ノード内での書き込みエリア重複は、初期化要求時にチェックし、リターンコードで報告します。

この場合の初期化要求は無効です。

(5) ソフト転写の機能

ソフト転写では、周期時間にデータを送信する周期送信型ソフト転写と、ネットワーク負荷を目的に転写周期の範囲内で分割型の送信をする分割送信型ソフト転写があります。

nx_ini_tmマクロのパラメータで設定することができます。

- ・周期送信型ソフト転写：m_typeにTMEM_SOFT
- ・分割送信型ソフト転写：m_typeにTMEM_FAST

<周期送信型ソフト転写>

①～④までのメッセージ（最大16320バイト）は排他制御によって整合性を保証します。
 最小転写周期は、100ミリ秒から設定できます。

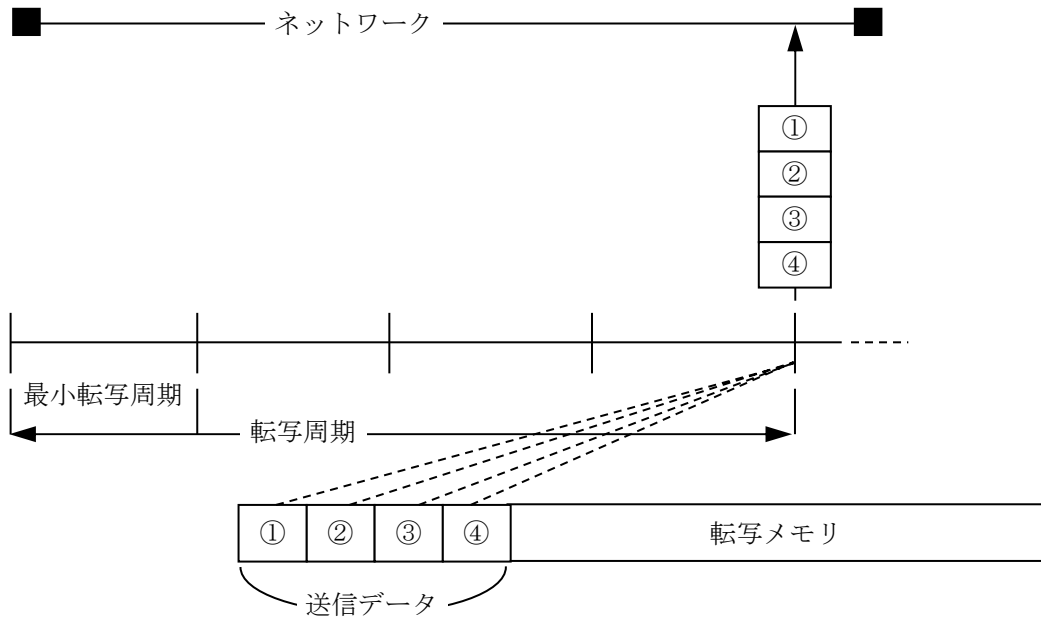


図 2-61 周期送信型ソフト転写の処理

<分割送信型ソフト転写>

排他制御はありません（2バイト境界のデータ保証となります）。
 最小転写周期は、10ミリ秒から設定できます。

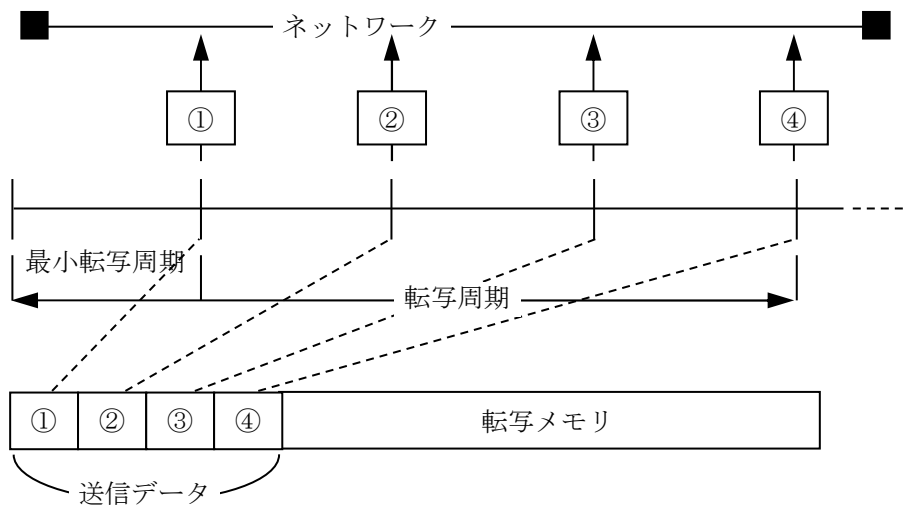


図 2-62 分割送信型ソフト転写の処理

分割送信型ソフト転写を使用した場合、周期送信型と分割送信型を受信することができます。

表 2-20 転写サポート機種一覧

	ハード転写	周期送信型ソフト転写	分割送信型ソフト転写
S10VE	—	○	○
S10V	○	○	○
S10mini	—	—	—
NX Dlinkを搭載した機種 (*)	○	○	○

○：サポート

—：未サポート（使用できません）

(*) NX Dlinkのサポート機種に関しては、「NX Dlink リファレンス（マニュアル番号 RS90-3-X051）」を参照してください。

なお、NX Dlinkでは、分割送信型ソフト転写を「高速ソフト転写」として記載していません。

(6) 最小転写周期の設定方法

最小転写周期は、毎周期に（1回の）送信実行が最も効率の良い設定となるため、下記の設定方法を推奨します。

【周期送信型ソフト転写】

転写データは一括送信となるため、次のようになります。

最小転写周期 = 転写周期

また、複数の転写をする場合の最小転写周期は、転写周期の最小公倍数を設定します。

【分割送信型ソフト転写】

高速型の場合、最小転写周期は、転写周期内に転写データを21ブロック単位に送信するための分割送信周期となります。

最小転写周期 = 転写周期 ÷ (送信ブロック数 ÷ 21)

└─ 小数点以下は切り捨て

└─ 小数点以下は繰り上げ

- ・複数の転写をする場合の最小転写周期は、上記で算出した値の最小公倍数を設定します。
- ・最小転写周期は、精度の問題上、最小でも10ms刻みとなるような設定（10ms、50ms、100msなど）を推奨します。

7. 3 構築時の留意事項

ソフト転写は、マルチキャスト通信の上位に搭載しているため、マルチキャスト通信を使用しない場合でも、マルチキャスト通信の諸定義データを設定する必要があります。

特に下記データについては、以下の点に留意してください。

【バッファ面数 (dfN : SBUFCNT、RBUFCNT)】

ソフト転写で使用するバッファ数も考慮して定義してください。

マルチキャスト通信では、通信ピーク時にサービスしきれず滞留するメッセージ数分をユーザが指定する必要がありますが、ソフト転写で使用するバッファ面数については、送信は書き込みエリア数分、受信は読み出しエリア数分確保する必要があります。

7. 4 マクロ概要

(1) 転写メモリマクロ一覧

下記マクロは、すべてTMIDを指定して発行します。つまり、TMID単位で制御できます。
 表2-21に転写メモリマクロの概略を示します。
 詳細は「第3編 マクロ仕様」を参照してください。

表2-21 転写マクロ一覧

マクロ名	概略機能	発行可能ステータス	補足
nx_init_tm	転写メモリの初期化	未定義状態	転写メモリを初期化します。
		停止状態	転写メモリの属性、転写メモリの制御情報を再設定します。
nx_ctl_tm	転写メモリの開始	停止状態または受信だけ実行中	送受信開始です。
	転写メモリの受信処理開始	停止状態	受信だけ開始です。
	転写メモリの停止	転写状態	送受信停止です。
nx_get_tm	転写メモリの制御情報リード	停止／転写状態	
nx_write_tm	転写メモリへの書き込み	停止／転写状態	未定義状態で発行した場合は、エラーリターンします。
nx_read_tm	転写メモリの読み込み	停止／転写状態	未定義状態で発行した場合は、エラーリターンします。

(2) 転写メモリの状態遷移

状態遷移は、nx_init_tm、nx_ctl_tmが正常終了した場合だけ有効です。
 また、nx_get_tm、nx_read_tm、nx_write_tmは、正常／異常を問わず状態遷移はありません。
 以下に転写メモリの状態遷移を示します。

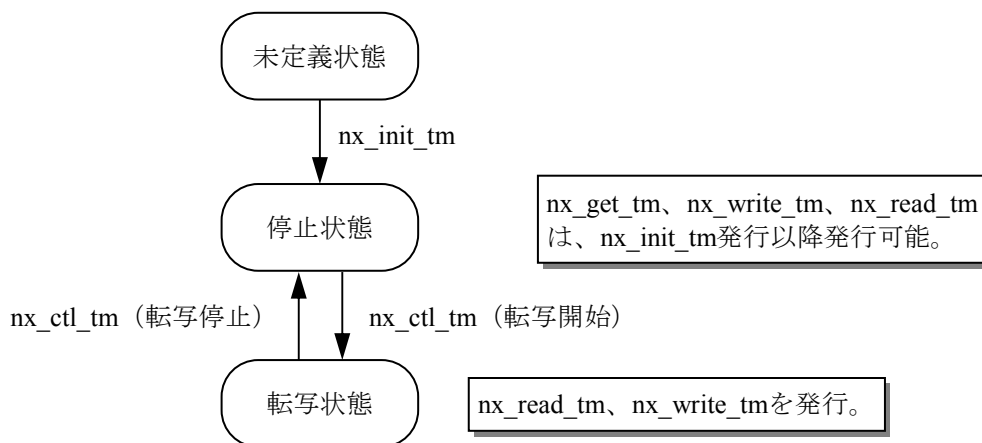


図2-63 転写メモリの状態遷移

【補足】

SIOVEのサイト構成は、マルチプロセッサ型（サイトはプロセッサごとに存在）であり、転写メモリの状態は、サイトごとに持っています。このため、各サイト内で転写に対する状態遷移を設定しなければなりません。

(3) データフィールドステータスとの関係

データフィールドのステータスは、CLOSEとOPENに分けられます。

転写メモリ機能は、データフィールドのステータスがOPENのときだけ使用できます。

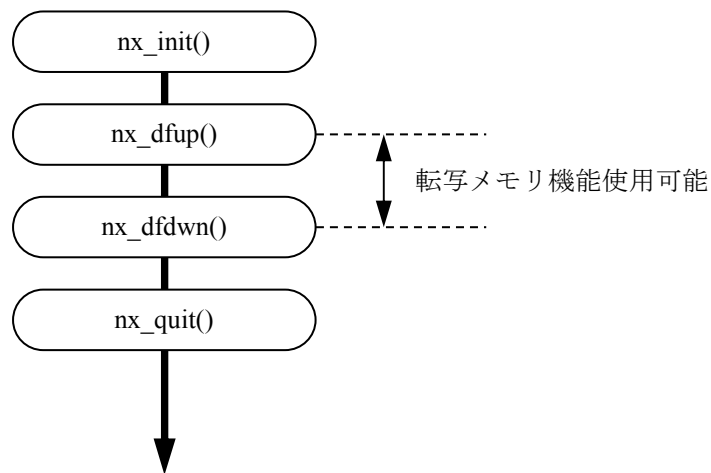


図 2-64 転写メモリとデータフィールド状態

第8章 システム構築機能

この章では、NXACPのインストールとシステム構築について、その手順を説明します。

NXACPは、通信制御をするタスク（以下、NXACPタスクと称します）とユーザインタフェース処理をするIRSUB（以下、NXACPマクロと称します）で動作し、オブジェクト本体はCD-ROMで提供します。

したがって、NXACP本体のインストールは、ターゲットとなるS10VEのサイト情報を所有する計算機からRPDPのコマンドを使用して実施します。

また、NXACP使用にあたり、動作環境を構築コマンドで設定する必要があります。

この環境設定も、ターゲットとなるS10VEのサイト情報を所有する開発系の計算機で実施し、S10VEのGLBエリアにローディング（以下、NXACPテーブルと称します）します。

以下にNXACPのシステム構築手順とそれに対応するコマンド名称を示します。なお、インストールおよびシステム構築は、すでにサイト情報の構築が終わっていることを前提としています。

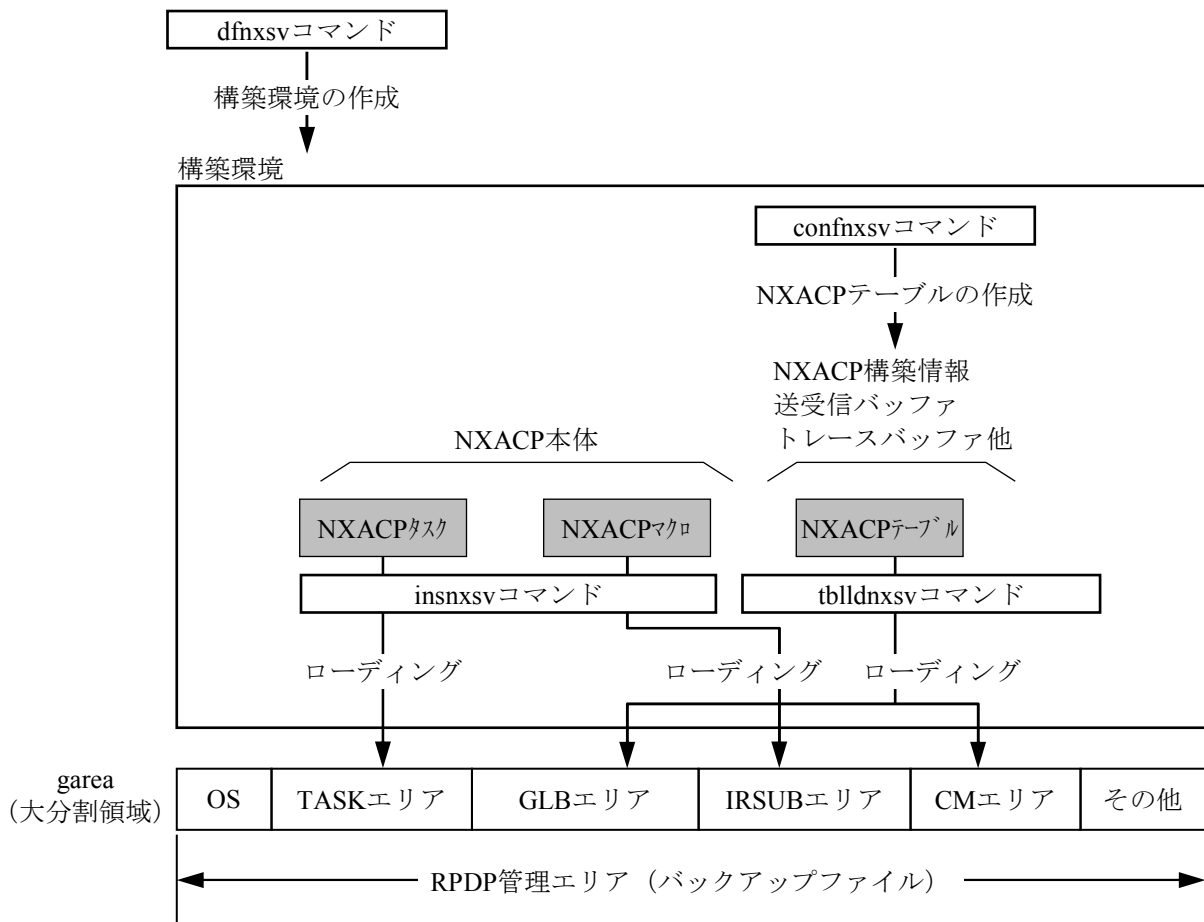


図 2-65 リソースの配置

【備考】 各構築コマンドは、実行中シグナルを抑止します。また、tblldnxsvは実行状態を表示します。

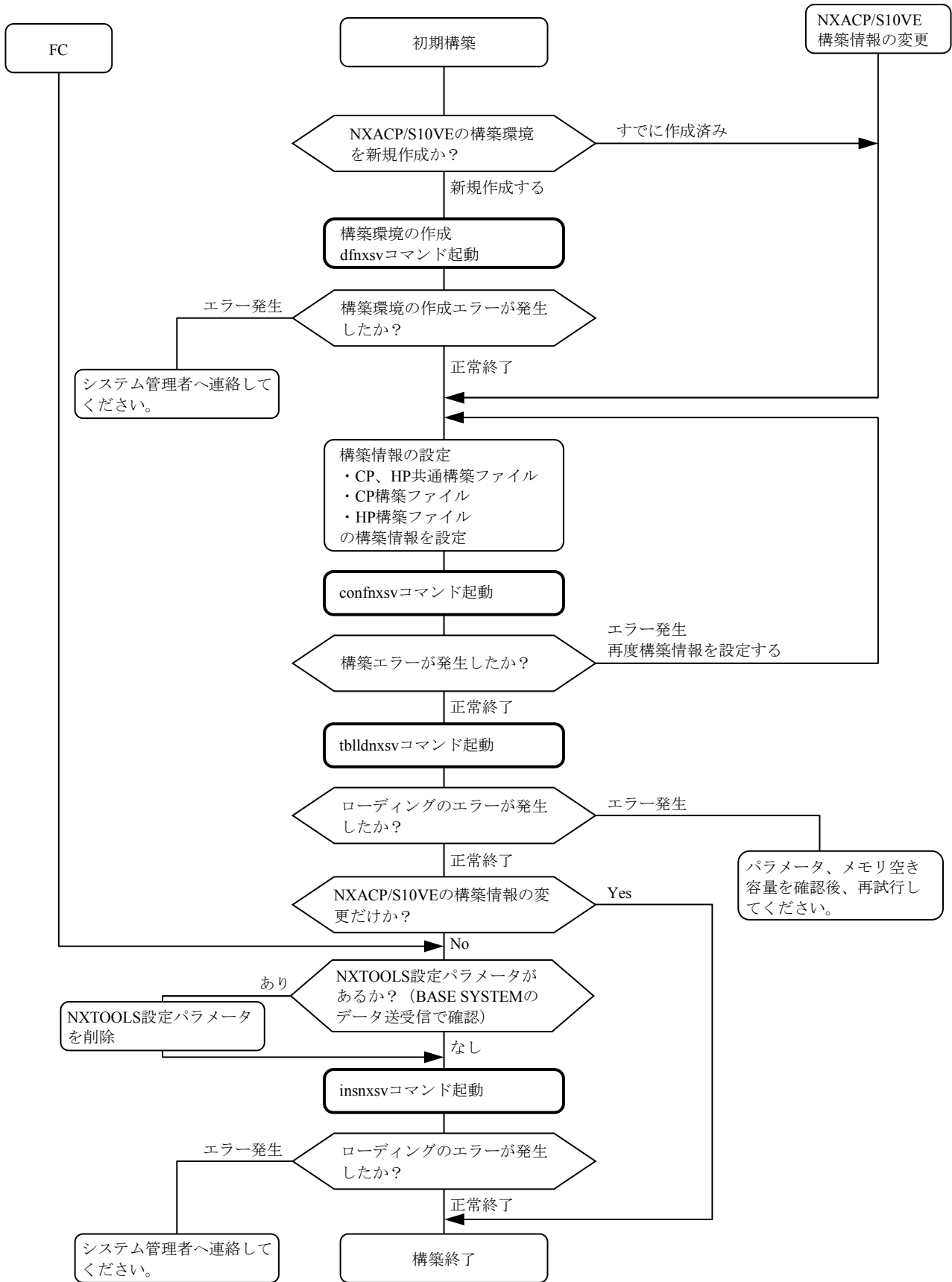


図 2-66 NXACPの構築手順

8. 1 NXACP本体のローディング

NXACPタスク、NXACPマクロなどのNXACP本体をサイト情報下のバックアップファイルへローディングするには、以下のコマンドを起動します。

```
# insnxsv サイト名称 [Enter]
```

指定するサイトは、CP側のサイト名称を指定してください。

このコマンドによって、サイト名称として指定したS10VEのバックアップファイルにNXACP本体がローディングされます。ターゲットとなるS10VEへの反映（ダウンロード）は、RPDPのリモートローディングコマンドを使用してください。

<エラー表示とその内容>

パラメータ数誤り：以下のエラーメッセージを表示します。

```
parameter number out of range  
usage:insnxsv site
```

実行中のエラー：RPDP関連のエラーを表示して処理を中断した場合、「S10VE ソフトウェア マニュアル オペレーション RPDP For Windows®（マニュアル番号 SEJ-3-133）」を参照してください。

<実行中の表示とその内容>

コマンドが正常に動作している場合、このコマンドは以下のメッセージを表示します。

```
*** NXACP/S10VE INSTALL START(CPU名称) ***  
          | ←————— CP側実行コマンドを表示します。  
*** CP(サイト名称) INSTALL OK !! ***  
          | ←————— HP側実行コマンドを表示します。  
*** HP(サイト名称) INSTALL OK !! ***  
*** NXACP/S10VE INSTALL END ***
```

【参考】 以下にNXACPのリソース割り付けを示します。

表2-22 リソース割り付け

プログラム名称	タスク番号	IRサブ番号	ULサブ番号	タスクレベル	格納エリア名称 (容量)
nx_memac	209	—	—	7	task/nxacpt (288KB)
nx_cycsnd	210	—	—	7	
(reserve)	211	—	—	—	
(reserve)	212	—	—	—	
nx_operation	213	—	—	4	
nx_snd1	214	—	—	6	
nx_snd2	215	—	—	6	
nx_snd3	216	—	—	6	
nx_snd4	217	—	—	6	
nx_snd5	218	—	—	6	
nx_snd6	219	—	—	6	
nx_htim	220	—	—	5	
nx_ltim	221	—	—	17	
nx_upexe	222	—	—	6	
nx_purcv (reserve)	223 224	— —	— —	6 —	
nx_init	—	301	—	—	
nx_quit	—	302	—	—	
nx_put	—	303	—	—	
nx_get	—	304	—	—	
nx_dfup	—	305	—	—	
nx_dfdwn	—	306	—	—	
nx_init_tm	—	307	—	—	
nx_ctl_tm	—	308	—	—	
nx_get_tm	—	309	—	—	
nx_write_tm	—	310	—	—	
nx_read_tm	—	311	—	—	
nx_trc	—	312	—	—	
nx_cdoff	—	313	—	—	
nx_cdon	—	314	—	—	
nx_puni (reserve)	— —	315 316~332	— —	— —	
(reserve)	—	—	2	—	
nx_ins	—	—	5 (INS)	—	
nx_exs	—	—	6 (EXS)	—	
nx_abs	—	—	7 (ABS)	—	
nx_ctl	—	—	15 (MODES)	—	

【補足】

タスク番号：209～224は、NXACPでリザーブしています。

IRサブ番号：301～332は、NXACPでリザーブしています。

ULサブ番号のENTRY No.：2は、NXACPでリザーブしています。

【留意事項】

NXACPを使用するユーザタスクは、NXACPタスクより低いタスクレベルを使用してください。

NXACPを使用するユーザタスクは、タスク番号を1～208の範囲内に設定してください。

【注意事項】

NXTOOLS SYSTEM/S10VEで設定したパラメータ情報が残っている場合、誤動作する場合があります。

ローディングする前にBASE SYSTEM/S10VEのデータ送受信機能で、NXTOOLS SYSTEM/S10VEで設定したパラメータ情報（“NX（タイプ4）”、“NX（タイプ5）”、または“NX（タイプ6）”）がある場合、削除してください。

BASE SYSTEM/S10VEの立ち上げ方法やデータ送受信機能の使用方法については、「S10VE ユーザーズマニュアル 総合編（マニュアル番号 SEJ-1-001）」を参照してください。

8. 2 システム構築とローディング

NXACPを使用するためには、サイト単位に動作環境の構築を行ってください。

8. 2. 1 初期設定

サイト単位の各定義情報は、以下のコマンドによって初期設定してください。なお、下記コマンドは、基本システムによりサイトを生成した後で初期構築時の1回だけ実行してください。

指定するサイトは、CP側のサイト名称を指定してください。

```
# dfnxsv サイト名称 [Enter]
```

<エラー表示とその内容>

パラメータ数誤り：以下のエラーメッセージを表示します。

```
parameter number out of range
```

```
usage:dfnxsv site
```

<構築環境>

サイトは、RPDPのサイト環境と同様にS10VEのプロセッサの構成上、CPU名の下にサイトをCP、HPごとに別けて作成します（なお、RPDPの仕様上、サイト名は、R700、R900のサイト名と重複しないユニークな名称（重複した場合、システムジェネレーションでエラー）となっていて、CPU名はサイト（CP）名になります）。

なお、R700、R900の構築環境と共存させるために、下記の構造となります。

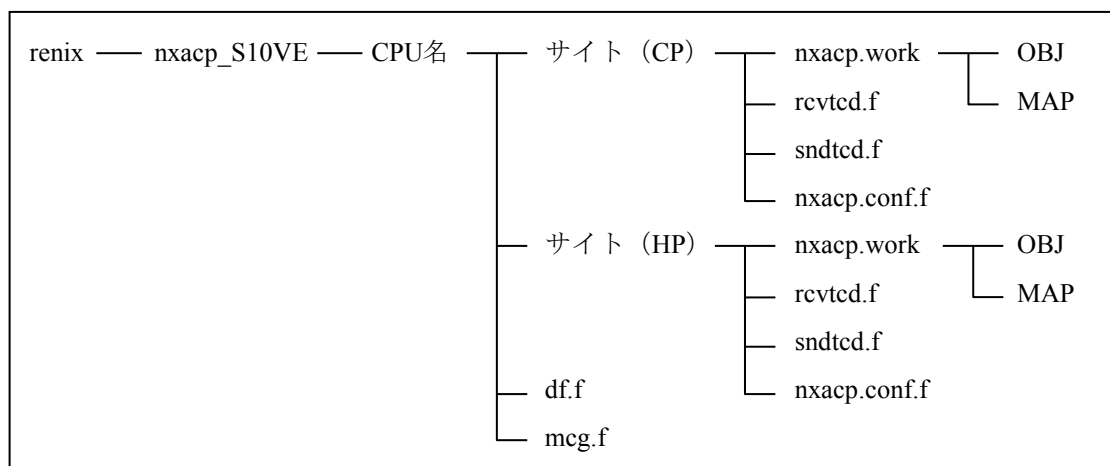


図 2-67 構築環境

8. 2. 2 構築情報の設定

サイト単位の各定義情報は、dfnxsvコマンドで指定サイト専用ディレクトリに格納された各定義「原本ファイル名称」を「作成ファイル名称」に変更した後、作成ファイルに対しエディタコマンド（notepad.exe）などで構築情報を設定してください。

表 2-23 構築ファイル一覧

分類	原本ファイル名称	作成ファイル名称	データベース管理内容
NXACP 共通定義	nxacp.conf.f	nxacp.conf	NXACPの基本的な設定値を管理します。
DF定義	df.f	dfN	データフィールドごとの設定値を管理し、ノードのLANアドレスなどの情報を定義してください。Nはデータフィールド番号を指定し、0~255まで指定できます。
MGN定義	mcg.f	mcgN	マルチキャストグループ番号ごとのオンライン/テスト用UDP宛先ポート番号を管理します。Nはデータフィールド番号を指定し、1~255まで指定できます。
TCD定義	rvctcd.f sndtvd.f	rvctcdN sndtcdN (*)	トランザクションコードごとの設定値を管理します。 rvctcdNは、TCDメッセージを受信するユーザタスク番号を定義します。Nはデータフィールド番号を指定し、0~255まで指定できます。なお、HP側で送受信をする場合、CP側で使用するしないに関係なくHP側の定義をCP側へ反映する必要があります。

(*) sndtcdNは、TCDメッセージを送信するユーザタスクを定義します。

ただし、このファイルは必須ではありません。後述する「TCDの送信権チェック」を使用する場合だけ作成してください。

各々の定義情報ファイルについて説明します。説明に先立って定義ファイルの記述方法を以下に示します。

定義ファイルは、以下の規則に従って作成してください。

<作成規則>

- 複数行定義の扱い
各定義情報は1行内で完結させ、複数行にまたがってはいけません（¥による行継続はできません）。
- 定義可能な文字
半角の英数字（数字は10進数）だけ定義可能とします。空白、タブ、空行は読み飛ばします。
行の先頭文字が#（シャープ）の場合、該当行をコメントと判断します。
記号、全角文字は定義不可能です。
- 定義順番
昇順、降順を問わずランダム可能とします。

(1) 共通定義ファイル (nxacp.conf) 設定情報

共通定義ファイルには、以下の情報を設定してください。

個々の設定が不要の場合は、「未使用時」の欄に示す数値を設定するか、認識子を削除、または認識子の行先頭に# (シャープ) を入れてください。なお、「未使用時」の欄が「-」の場合、省略できません。

認識子	内容	指定可能範囲	未使用時
MAXDFCNT	登録データフィールド総数	0~16 (*1)	-
TRCCS	トレースケース数 (ケース数)	0、128~65535 (128の倍数)	0
TMSCYC	ソフト転写動作周期 (10ミリ秒の倍数で指定。ソフト転写未使用は10指定)	10~8540000 (10の倍数)	1000
DFNO	データフィールド番号	0~255 (*2)	-

(*1) HP側のnxacp.confの定義情報「MAXDFCNT」(登録データフィールド総数)に「0」(使用するデータフィールドなし)を設定してください。CP側では、MAXDFCNTを「0」に設定できません。

(*2) 定義したデータフィールド番号で後述する「dfN」を作成する必要があります。

下記の<定義例>では、「DFNO 0」に対して「df0」、「DFNO 1」に対して「df1」を作成する必要があります。

<定義例>

MAXDFCNT	2
TRCCS	512
TMSCYC	100
DFNO	0
DFNO	1

(2) データフィールド定義ファイル (dfN) 設定情報

データフィールド定義ファイルには、以下の情報を設定してください。

右端のI/L/Rは、データフィールドのタイプ (I: 自ノード内、L: ローカル、R: リモート) と指定の有無 (○: 指定要、×: 不要 (指定無効)、△: ローカルDFの定義に従う) を示しています。

個々の設定が不要の場合は、「未使用時」の欄に示す数値を設定するか、認識子を削除、または認識子の行先頭に# (シャープ) を入れてください。なお、「未使用時」の欄が「-」の場合、省略できません。

認識子	内容	指定可能範囲	未使用時	L	R	I
DFTYPE	データフィールドタイプ (=0: ローカル、≠0: リモート (ソースデータフィールド番号))	0、1~255	-	○	○	×
NODENAME	自ノード名称	ASCII9文字以内	-	○	×	×
NODENO	自ノード番号	1~4095	-	○	×	×
SSPORT	送信用自ポート番号	1~65535	-	○	△	×
ALPORT	生存信号宛先ポート番号	1~65535	-	○	×	×
ALSCYC	生存信号送信周期 (秒)	1~3600	-	○	×	×
ALTOUT	生存信号タイムアウト時間 (秒)	1~3600	-	○	×	×
ALRCVFG	生存信号受信要否フラグ (0: 受信しない、1: 受信する)	0、1	0	○	×	×
UNO1	LAN1のUNO	1~23	-	○	×	×
UNO2	LAN2のUNO	2~23	0	○	×	×
MINSNDMCG	最小送信マルチキャストグループ番号	1~255	0	○	○	×
MAXSNDMCG	最大送信マルチキャストグループ番号	1~255	0	○	○	×
SNDMCGCNT	送信マルチキャストグループ総数	1~128	0	○	○	×
MINRCVMCG	最小受信マルチキャストグループ番号	1~255	0	○	×	×
MAXRCVMCG	最大送信マルチキャストグループ番号	1~255	0	○	×	×
RCVMCGCNT	受信マルチキャストグループ総数	1~32	0	○	×	×
MINUSETCD	最小送受信TCD番号	1~59999	-	○	○	○
MAXUSETCD	最大送受信TCD番号	1~59999	-	○	○	○
USETCDCNT	送受信TCD総数	1~3328	-	○	○	○
SBUFSZ	送信バッファ1ケースサイズ	512~1408	0	○	△	×
SBUFCNT	送信バッファケース数	1~1024	0	○	○	×
RBUFSZ	受信バッファ1ケースサイズ	512~1408	0	○	△	○
RBUFCNT	受信バッファケース数	1~1024	0	○	○	○
BUFHWRATE	HIGH WATERアラームバッファ使用率	2~100	-	○	○	○
BUFLWRATE	LOW WATERアラームバッファ使用率	1~99	-	○	○	○
MINNODENO	最小ノード番号	1~4095	0	○	○	×
MAXNODENO	最大ノード番号	1~4095	0	○	○	×
R_INA1	LAN1ネットワークアドレス	XX.XX.XX.XX	-	×	○	×
R_NMASK1	LAN1ネットマスク	XX.XX.XX.XX	-	×	○	×
R_INA2	LAN2ネットワークアドレス	XX.XX.XX.XX	0	×	○	×
R_NMASK2	LAN2ネットマスク	XX.XX.XX.XX	0	×	○	×
CYCFG	ネットワーク共有メモリ使用フラグ (0: 使用しない、1: 使用する)	0、1	0	○	×	×
SYSPORT	システム用マルチキャスト通信ポート番号	1~65535	0	○	○	×
UNIMCGNO	送信ユニキャストグループ番号	MINSNDMSG ~MAXSNDMSG	0	○	○	×

<メモリの有効利用—口メモ>

受信するTCD、MGNは、ノードごとに連続して使用することを推奨します。

また、ノード番号は、データフィールドごとに連続して使用することを推奨します。

【留意事項】

各認識子の留意事項を以下に示します。

SBUFSZ、

RBUFSZ：指定バイト数は、バッファ1ケースサイズ中のユーザデータサイズで4の倍数で指定してください。

実際のバッファ1ケースサイズは、ここでの指定値に64バイト（NXヘッダ）と4バイト（バッファ管理ヘッダ）を加えたサイズです。

MINUSETCD、

MAXUSETCD、

USETCDCNT：送受信TCDを定義してください。ただし、送信TCDチェックを行わない場合、受信についてだけ定義してください。

R_INA：リモートデータフィールド用のデータです。10進の.（ドット表記）で指定してください。

R_NMASK：リモートデータフィールド用のデータです。10進で指定してください。

CYCFG：ネットワーク共有メモリの詳細データはマクロで設定してください。

SBUFCNT、

RBUFCNT：自ノード内データフィールドでRBUFCNT=0の指定はできません。

リモートデータフィールドでSBUFCNT=RBUFCNTに0を指定してもエラーとはしませんが、実際の送受信は不可となります。

ローカルデータフィールドでSBUFCNT=RBUFCNT=0を指定した場合は、生存信号の送信だけとなります。

【留意事項】の続き

ALSCYC、

ALSTOUT : この2つのパラメータは「ALSCYC<ALSTOUT」の関係になるよう設定してください。

<ローカルデータフィールド定義例>

DFTYPE	0
NODENAME	OOMIKA
NODENO	3
SSPORT	1025
ALPORT	600
ALSCYC	10
ALSTOUT	30
ALRCVFG	0
UNO1	3
UNO2	0
MINSNDMCG	1
MAXSNDMCG	4
SNDMCGCNT	4
MINRCVMCG	5
MAXRCVMCG	8
RCVMCGCNT	4
MINUSETCD	1
MAXUSETCD	64
USETCDCNT	32
SBUFSZ	1408
SBUFCNT	128
RBUFSZ	1408
RBUFCNT	128
BUFHWRATE	80
BUFLWRATE	30
MINNODENO	1
MAXNODENO	64
CYCFG	0
SYSPORT	0

<自ノード内通信データフィールド定義例>

DFTYPE	0
MINUSETCD	1
MAXUSETCD	64
USETCDCNT	32
RBUFSZ	1408
RBUFCNT	128
BUFHWRATE	80
BUFLWRATE	30

<リモートデータフィールド定義例>

DFTYPE	1
MINSNDMCG	1
MAXSNDMCG	4
SNDMCGCNT	4
SBUFCNT	128
RBUFCNT	128
BUFHWRATE	80
BUFLWRATE	30
MINNODENO	1
MAXNODENO	64
R_INA1	192.168.0.1
R_NMASK1	255.255.255.0
R_INA2	0
R_NMASK2	0

(3) マルチキャストグループ定義ファイル (mcgN) 設定情報

このファイルには、以下の情報を設定してください。

右端のI/L/Rは、データフィールドのタイプ (I: 自ノード内、L: ローカル、R: リモート) と指定の有無 (○: 指定要、×: 不要 (指定無効)、△: ローカルDFの定義に従う) を示しています。

個々の設定が不要の場合は、「未使用時」の欄に示す数値を設定するか、認識子を削除、または認識子の行先頭に# (シャープ) を入れてください。なお、「未使用時」の欄が「-」の場合、省略できません。

認識子	内容	指定可能範囲	未使用時	L	R	I
SMGN	送信マルチキャストグループ番号	1~255 (*1)	-	○	○	×
RMGN	受信マルチキャストグループ番号	1~255 (*1)	-	○	×	×
OPORT	オンライン用ポート番号 (*2)	1~65535	0	○	○	×
TPORT	テスト用ポート番号 (*2)	1~65535	0	○	○	×

(*1) 1送受信マルチキャストグループの登録個数は、dfNに登録したSNDMCGCNT、RCVMCGCNTまでが有効です。

(*2) 送信時は宛先送信ポート番号で、受信時はbind()するポート番号です。オンライン用/テスト用のポート番号は、送受信別に定義してください。

<定義例>

#MGNTYPE	MGN	OPORT	TPORT
SMGN	1	55001	57001
SMGN	2	55002	57002
SMGN	3	55003	57003
SMGN	4	55004	57004
#MGNTYPE	MGN	OPORT	TPORT
RMGN	5	55005	57005
RMGN	6	55006	57006
RMGN	7	55007	57007
RMGN	8	55008	57008

オンラインまたはテストモードだけ使用したい場合は、未使用のモードに対するポート番号に0を指定してください (ただし、ポート番号に0を指定したノードモードでnx_dfup()を発行した場合は、ソケットマクロエラーで異常終了します)。

【留意事項】

- 受信マルチキャストグループに自ノードが送信するマルチキャストグループ番号を指定した場合でも、自ノードが送信するメッセージは受信できませんので注意してください。
- dfNのSSPORT、ALPORT、SYSPORTで定義したポート番号をmcgNに定義しないでください。
- テストモードのポートを使用して、NX Dlink (WS、PC、制御サーバ用NX通信パッケージ) と通信する場合、NX DlinkのdfieldN定義情報内LNSYSTYPEは「0」を指定してください。
- NX Dlinkと通信する場合、NX DlinkのdfieldN定義情報内RCVLEVELは「1」を指定してください。

(4) 受信トランザクションコード定義ファイル (rcvtcdN) 設定情報

このファイルには、以下の情報を設定してください。

右端のI/L/Rは、データフィールドのタイプ (I: 自ノード内、L: ローカル、R: リモート) と指定の有無 (○: 指定要、×: 不要 (指定無効)、△: ローカルDFの定義に従う) を示しています。

個々の設定が不要の場合は、「未使用時」の欄に示す数値を設定するか、認識子を削除、または認識子の行先頭に# (シャープ) を入れてください。なお、「未使用時」の欄が「-」の場合、省略できません。

認識子	内容	指定可能範囲	未使用時	L	R	I
TCDNO	受信トランザクションコード番号	1~59999 (*1)	-	○	×	○
BUFCNT	最大受信バッファケース	4~1024 (*2)	-	○	×	○
TN	受信タスク番号	1~208 (*3)	-	○	×	○

(*1) TCDNOの登録個数は、dfNに登録したUSETCDCNTまでです。

また、TCDNOの指定範囲は、dfNに登録したMINUSETCDからMAXUSETCDまでです。

(*2) 最大バッファケースは、dfNに登録したRBUFCNTまでです。0指定時は、dfNのRBUFCNTの値となります。また、接続リモートDFがある場合、リモートDF用のdfNのRBUFCNTを加えた値となります。

(*3) 1つのTCDNOに対して登録できるタスク数は、最大8タスクです。

<定義例>

#RCV	TCDNO	BUFCNT	TN1	TN2	TN3	TN4	TN5	TN6	TN7	TN8
TCD	1	16	1	11	21	31	41	71		
TCD	2	16	2	112	142					
TCD	3	16	16	32	48	64	80	96	112	128
TCD	4	16	64	128						

(5) 送信トランザクションコード定義ファイル (sndtcdN) 設定情報

このファイルには、以下の情報を設定してください。ただし、TCD送信権チェックが不要の場合、このファイルを作成する必要はありません。

右端のI/L/Rは、データフィールドのタイプ (I: 自ノード内、L: ローカル、R: リモート) と指定の有無 (○: 指定要、×: 不要 (指定無効)、△: ローカルDFの定義に従う) を示しています。

個々の設定が不要の場合は、「未使用時」の欄に示す数値を設定するか、認識子を削除、または認識子の行先頭に# (シャープ) を入れてください。なお、「未使用時」の欄が「-」の場合、省略できません。

認識子	内容	指定可能範囲	未使用時	L	R	I
TCDNO	送信トランザクションコード番号	1~59999 (*1)	-	○	○	○
TN	送信タスク番号	1~208 (*2)	-	○	○	○

(*1) TCDNOの登録個数は、dfNに登録したUSETCDCNTまでです。

また、TCDNOの指定範囲は、dfNに登録したMINUSETCDからMAXUSETCDまでです。

(*2) 1つのTCDNOに対して登録できるタスク数は、最大8タスクです。

<定義例>

#SND	TCDNO	TN1	TN2	TN3	TN4	TN5	TN6	TN7	TN8
TCD	1	1	11	21	31	41	71		
TCD	2	2	112	142					
TCD	3	16	32	48	64	80	96	112	128
TCD	4	64	128						

【留意事項】

登録したタスクは、送信する全データフィールドに対して送信TCDとの対応付けをする必要があります。データフィールド単位に送信チェックの有無を指定することはできません。

8. 2. 3 構築情報のコンパイル

サイト単位の各定義情報は、以下のコマンドでコンパイルしてください。

なお、下記コマンドは、構築情報変更時も必ず実行してください。

```
# confnxsv サイト名称 [Enter]
```

<エラー表示とその内容>

パラメータ数誤り：以下のエラーメッセージを表示します。

```
parameter number out of range
```

```
usage:confnxsv site
```

指定するサイトは、CP側のサイト名称を指定してください。

なお、confnxsvコマンド実行中にコンパイラなどのエラーメッセージが出力された場合は、構築情報を見直してください。HP側構築情報のコンパイルで異常となった場合には、HP側のnxacp.confの定義情報「MAXDFCNT」に「0」を設定してください。

<実行中の表示とその内容>

コマンドが正常に動作している場合、このコマンドは以下のメッセージを表示します。

```
*** NXACP/S10VE GENERATION START(CPU名称) ***
```

```
          | ← CP側構築情報をチェック
```

```
*** CP(サイト名称) CONFIGURATION OK !! ***
```

```
          | ← HP側構築情報をチェック
```

```
*** HP(サイト名称) CONFIGURATION OK !! ***
```

```
*** NXACP/S10VE GENERATION END ***
```

【補足】

S10VEでは、まったく同じ構築情報を他ノードで使用する場合、構築ファイルを一式コピーした後、ノード名称、ノード番号を変更し（dfNを修正）、上記コマンドを起動すれば簡単にサイト情報の複製ができます（構築環境のOBJディレクトリ下にあるオブジェクトファイルをすべて消去してから行ってください）。

8. 2. 4 構築情報のローディング

confnxsvコマンドで出力された情報を、tblldnxsvコマンドで指定サイトに制御テーブルとしてローディングします。

```
# tblldnxsv サイト名称 ON/OFF [Enter]
```

このコマンドによって、構築情報を実機にローディングします。

ONオプションを指定した場合、サイト名称として指定したS10VE実機にNXACPの構築情報をローディングします。このとき、NXACPを一度強制停止させますので注意してください。

OFFオプションを指定した場合、サイト名称として指定したS10VEのメモリイメージファイルにNXACPの構築情報がローディングされます。S10VE実機へのローディングは、RPDPのリモートローディングコマンド (svrpl) を使用して一括ローディングしてください。

<エラー表示とその内容>

パラメータ数誤り：以下のエラーメッセージを表示します。

```
parameter number out of range
usage:tblldnxsv site ON/OFF
```

実行中のエラー：RPDP関連のエラーを表示して処理を中断した場合、「S10VE ソフトウェアマニュアル RPDP For Windows® (マニュアル番号 SEJ-3-133)」を参照してください。

<実行中の表示とその内容>

コマンドが正常に動作している場合、このコマンドは以下のメッセージを表示します。

```
*** NXACP/S10VE TABLE LOAD START(CPU名称) ***
          |           ← CP側テーブルロードをチェック
*** CP(サイト名称) TABLE LOAD OK !! ***
          |           ← HP側テーブルロードをチェック
*** HP(サイト名称) TABLE LOAD OK !! ***
*** NXACP/S10VE TABLE LOAD END ***
```

【補足】

構築ファイルを一式コピーした後でtblldnxsvコマンドを起動する場合、OBJディレクトリ下の.objファイルをすべて消去してから起動してください (tblldnxsvコマンドでは、.objファイルをコンペアするため、新規に構築ファイルをコピーした場合、サイトバックアップファイルにロードされていると判断してしまうのでテーブルが正常にロードされません)。

8.3 容量見積り

NXACPでは、各種最大値、テーブル数などを様々なシステム形態に適合させるためにシステム構成情報をパラメータ化しています。パラメータの指定にあたって、NXACPを構築するために必要なメモリ容量の算出方式を示します。なお、以下に示す認識子は、構築情報の認識子です。

また、実際の確保容量は、プロセッサごとの合計容量をページ単位（4096バイト）で割り付けることとなりますので注意してください。

表2-24 容量算出シート

エリア名称	1ケースバイト数 (対応構築ファイル名)	ケース数 (対応構築ファイル名)	合計 (*2) (バイト)
共通固定部			53,248
自 ノ ード 内	受信バッファ	$RBUFSZ(df0)+228$	$RBUFCNT(df0)$
	TCD管理部	4	$MAXUSETCD(df0) - MINUSETCD(df0) + 1$
		64	$USETCDCNT(df0)$
ロ ー カ ル D F	送信バッファ	$SBUFSZ(dfN)+100$ (*3)	$SBUFCNT(dfN)$
	受信バッファ	$RBUFSZ(dfN)+228$ (*4)	$RBUFCNT(dfN)$
	TCD管理部	4	$MAXUSETCD(dfN) - MINUSETCD(dfN) + 1$
		64	$USETCDCNT(dfN)$
	送信MCG管理部	16	$MAXSNDMCG(dfN) - MINSNDMCG(dfN) + 1$
	受信MCG管理部	1	$MAXRCVMCG(dfN) - MINRCVMCG(dfN) + 1$
		32	$RCVMCGCNT(dfN)$
	ノード管理部	32	$MAXNODENO(dfN) - MINNODENO(dfN) + 1$
32		$RCVMCGCNT(dfN) \times$ 定義ノード数 (*1)	
リ モ ー ト D F	送信バッファ	$SBUFSZ$ (*5) +100 (*3)	$SBUFCNT(dfN)$
	受信バッファ	$RBUFSZ$ (*5) +228 (*4)	$RBUFCNT(dfN)$
	TCD管理部	4	$MAXUSETCD(dfN) - MINUSETCD(dfN) + 1$
		64	$USETCDCNT(dfN)$
	送信MCG管理部	16	$MAXSNDMCG(dfN) - MINSNDMCG(dfN) + 1$
	ノード管理部	32	$MAXNODENO(dfN) - MINNODENO(dfN) + 1$
32		$RCVMCGCNT$ (*3) \times 定義ノード数 (*1)	
トレース部	32	TRCCS	
合計			

(*1) 定義ノード数とは、 $MAXNODENO(dfN) - MINNODENO(dfN) + 1$ です。

(*2) 合計は、1ケースバイト数 \times ケース数の値です。

(*3) 100とは、送信バッファの管理情報のサイズです。

(*4) 228とは、受信バッファの管理情報のサイズです。

(*5) dfN 内DFTYPEで指定したソースDFの値です。

ローカルデータフィールドで、メモリ転写使用時は、上記で算出する数値に768バイト、また生存信号受信時は、上記で算出する数値に1536バイト加えたサイズがテーブル容量となります。

このページは白紙です。

第3編 マクロ仕様

第1章 総説

1. 1 マクロの種類と一覧

NXACPの提供するマクロは、IRSUB形態ですので、ユーザタスクでスタックサイズを確保してください。また、リターンコードは、数値を直接返しますので、`geterrno()`で読み込む必要はありません。

NXACPのマクロ一覧を表3-1に示します。

表3-1 マクロ一覧

<マルチキャスト通信マクロ>

マクロ名称	機能概要	スタックサイズ (バイト)	発行有無	
			CP	HP
<code>nx_put</code>	マルチキャストメッセージ送信	2048	○	×
<code>nx_get</code>	マルチキャストメッセージ送信	4096	○	×

<運用マクロ>

マクロ名称	機能概要	スタックサイズ (バイト)	発行有無	
			CP	HP
<code>nx_init</code>	NXACPの起動	4096	○	×
<code>nx_dfup</code>	データフィールドの起動	4096	○	×
<code>nx_dfdown</code>	データフィールドの停止	4096	○	×
<code>nx_quit</code>	NXACPの停止	4096	○	×

<共有メモリマクロ>

マクロ名称	機能概要	スタックサイズ (バイト)	発行有無	
			CP	HP
<code>nx_init_tm</code>	転写メモリの初期化	512	○	×
<code>nx_ctl_tm</code>	転写メモリの転写開始・停止	512	○	×
<code>nx_get_tm</code>	転写メモリの制御情報読み出し	512	○	×
<code>nx_write_tm</code>	転写メモリのデータ書き込み	512	○	×
<code>nx_read_tm</code>	転写メモリのデータ読み出し	512	○	×

○：発行可

×：発行不可

【留意事項】

- マクロインタフェースにおいて、現状未使用部分は、将来何らかの機能追加によって意味を持つ場合がありますので、必ず「0」を指定してください。
- NXACPのマクロは、ユーザタスクからだけアクセスできます。ユーザタスクとは「RSUTYP=u」で作成したタスクを指します。

第2章 マルチキャスト通信マクロ

2.1 nx_put

【名前】

nx_put - 1メッセージの送信

【形式】

```
#include <nxacp.h>
long nx_put (msg, msglen, ginfo);

char *msg;
long *msglen;
nx_ginfo *ginfo;
```

【パラメータ】

msg[in] 送信すべきユーザデータの先頭アドレスを指定します。

msglen[in] 送信すべきユーザデータのサイズ（バイト単位）を指定します。

nx_ginfo[in] 以下の送信情報を指定します。

```
typedef struct {
    unsigned char dfn; /* 送信先データフィールド番号 */
    char fu1; /* 将来用 (0固定) */
    unsigned short mgn; /* 送信先マルチキャストグループ番号 */
    unsigned short tcd; /* 送信トランザクションコード */
    short fu2; /* 将来用 (0固定) */
    unsigned long sa; /* nx_get()用のエリアでnx_put()では未使用 (0固定) */
    long revlen; /* nx_get()用のエリアでnx_put()では未使用 (0固定) */
}nx_ginfo;
```

【機能説明】

nx_put()は、マルチキャスト一方向通信だけをサポートし、1回の発行によって1つのメッセージを1つのデータフィールド内にブロードキャスト送信します。

送信メッセージエリアの先頭アドレスを「msg」に指定し、送信データバイト数を「msglen」に指定します。1回の発行で送信できるデータサイズは、16KBまでです。

自ノード宛にメッセージを送りたい場合には、dfnに宛先データフィールドとして「0」を指定してください。NXACPは、ネットワークへは送出することなく、NXACP内部で折り返します。他ノード宛にメッセージを送りたいときには、dfnに「0」以外を指定します。この場合、NXACPは自ノードへは折り返しません。

したがって、1回のnx_put()発行で1つのメッセージを自ノードと他ノードに送信することはできません。同じメッセージを自ノードと他ノードに送信したい場合には、宛先データフィールド番号を変えてnx_put()を複数回発行してください。

nx_put()は、NXACP内部のキューに送信データを格納した時点でリターンします。

nx_put()は、ノンブロックだけをサポートします。キュー満杯などの原因でキュー確保に失敗した場合には、キュー確保リトライは行わずそのままエラーリターンします。

送信メッセージの優先レベル、データのモードは、NXACPが判別し制御をします。

【戻り値】

nx_putは正常終了時、値0を返します。

異常終了の場合は、次の値を返します。

0x001：NXACPが初期化されていません。

0x003：構築情報がロードされていません。

0x102：dfnのデータフィールドが定義されていません。

0x103：dfnのデータフィールドが初期化されていません。

0x108：dfnに指定した他PUのデータフィールドが初期化されていません（状態遷移中含む）。

0x111：mgnの値が指定可能範囲（1～255）外です。

0x112：mgnのマルチキャストグループが定義されていません。

0x121：tcdの値が指定可能範囲（1～59999）外か定義されていません。

0x131：msglenが指定可能範囲（0～16384）外です。

0x201：要求に対する送信バッファを確保できませんでした。

0x211：全ネットワークモジュールに障害が発生しています。

0x231：PU間通信で異常を検出しました。

0x301：要求発行タスク番号が範囲（1～208）外です。

0x303：tcdの値が定義されていません（TCD送信権チェック指定時だけ）。

【補足】

送信宛先として指定するデータフィールドの情報およびマルチキャストグループの情報を、あらかじめ構築時にそれぞれデータフィールド構築情報、マルチキャストグループ構築情報として設定してください。

nx_put()のエラーコードで返すのは、NXACP内部の送信キュー接続までの異常情報だけです。

内蔵Etherでの障害発生については、EASでの障害通知になります。また、nx_put()が正常終了しても、送信処理前にネットワーク障害が発生したり、nx_quit()またはnx_dfdwn()で処理停止した場合は、ネットワークに送信されない場合もあります。なお、ネットワーク送信不可に陥った場合、送信キューに接続された送信メッセージは、NXACPで廃棄されます。

NXACPは、「同じタスクレベルからの送信要求順に処理し、その順序を保証してネットワークに送出」します。もし、順序を厳密に保証したい場合には、nx_put()がリターンしたことを確認してから次のデータの送信要求nx_put()を発行してください。

2. 2 nx_get

【名前】

nx_get — 1メッセージの受信

【形式】

```
#include <nxacp.h>

int nx_get (buf, buflen, offset, nx_ginfo, time);

char *buf;
long *buflen,*offset;
nx_ginfo *ginfo;
long *time;
```

【パラメータ】

buf[out] 受信データ格納エリアの先頭アドレスを指定します。このエリアに受信したデータが格納されます。

buflen[in] 受信データ格納エリアサイズを指定します。
到着メッセージがこのエリアサイズより大きい場合、受信要求サイズとして処理されます。

offset[in] メッセージの切り出し受信の際の、切り出すメッセージの相対アドレス（先頭を0とした相対アドレス）を指定します。メッセージ切り出し受信を行わない場合には、「0」を指定してください。

nx_ginfo[out] 以下の送信情報を指定します。

```
typedef struct {
    unsigned char dfn; /* 受信データフィールド番号 */
    char fu1; /* 将来用 (0固定) */
    unsigned short mgn; /* 受信マルチキャストグループ番号 */
    unsigned short tcd; /* 受信トランザクションコード */
    short fu2; /* 将来用 (0固定) */
    unsigned long sa; /* 送信元アドレス (*) */
    long rcvlen; /* 受信したデータサイズ。 */
    /* ただし、タイムアウト時は0xFFFFFFFF。 */
}nx_ginfo;
```

time[in] 受信タイムアウト監視時間（秒）を指定します。

(*) 送信元アドレスは、「付録F メッセージヘッダフォーマット」のSAと同じ内容です。

【機能説明】

`nx_get()`は、構築時に指定したTCD付きメッセージを1つ受信します。

受信データ格納エリアの先頭アドレスを「`buf`」に指定し、エリアサイズを「`buflen`」に指定します。

また、メッセージの切り出し受信を行いたい場合、切り出すメッセージの相対アドレス（先頭を0とした相対アドレス）を「`offset`」に指定します。

データ受信時は、`buf`には受信データが格納され、`rcvlen`に受信データサイズが格納されます。

このとき、受信データは`msg_offset`で切り出されたものが格納されます。`buflen`よりも到着したメッセージのサイズが大きい場合には、`offset`で指定された相対アドレスから`buflen`サイズ分が切り出され`buf`に格納されます。

1回の`nx_get()`で受信できるユーザデータサイズは、16KBまでです。

受信データの送信元データフィールド番号、送信元ノード番号、宛先マルチキャストグループ、トランザクションコードなどの詳細情報は、`nx_ginfo`に格納されます。

`time`は、データ到着を待つ時間です。指定可能時間は、-1~3600（秒）です（秒単位）。

ただし、「0」指定は、メッセージ到着を無限に待つモードとなります。

また、受信メッセージが未到着の場合でメッセージ到着を待ちたくない場合は、「-1」を指定してください。「-1」を指定した場合、`nx_get()`は、メッセージが到着していれば受信処理をし、メッセージが到着していなければ戻り値に「0」を設定し、`rcvlen`に「-1」を設定して処理を終了します。

タイムアウトした場合でも、NXACPでは正常として扱い、戻り値に「0」を設定してユーザに制御を戻しますが、`rcvlen`には「-1」を設定して戻します。

ユーザタスクがどのメッセージを受信するかは構築時に指定しますので、`nx_get()`でその中の特定のメッセージを指定して受信することはできません。

ネットワークから受信した順にユーザタスクキューに接続し、先入し先出し（FIFO）でユーザタスクにサービスされます。

【戻り値】

`nx_get`は正常終了時、値0を返し、`rcvlen`に受信したデータサイズ（0以上）を設定します。ただし、データ到着待ち時間超過による処理終了の場合は、`rcvlen`に「-1」を設定します。異常終了の場合は、次の値を返します。

0x001 : NXACP初期化されていない（`nx_init()`未発行）か停止（`nx_quit()`発行）されました。

このコードは、制御情報入れ替え時も発生します。

0x003 : 構築情報がロードされていません。

0x131 : `buflen`に負の値が設定されています。

0x141 : `offset`が指定可能範囲（0~16384）外です。

0x151 : `time`が指定可能範囲（0~3600、-1）外です。

0x202 : `nx_dfdown()`発行によって受信バッファがページされました。

0x301 : 要求発行タスク番号が範囲（1~208）外です

0x302 : 発行元ユーザタスクの受信TCDが定義されていません。

【補足】

受信したいメッセージのTCDは、構築時にあらかじめ設定しておいてください。

`nx_get()`のエラーコードで返すのは、NXACP内部の受信キューからのメッセージ取り出しまでの異常情報だけです。内蔵Etherでの障害発生については、EASでの障害通知になります。

データフィールドを`nx_dfdown()`で停止した場合または`nx_quit()`でNXACPを停止した場合、処理待ちメッセージはすべてページされます。

また、`nx_quit()`を発行した場合、`nx_get()`は処理を中断しユーザに制御を戻します。

第3章 運用マクロ

3.1 nx_init

【名前】

nx_init — NXACPの立ち上げ

【形式】

```
#include <nxacp.h>
```

```
long nx_init (info)
```

```
long info[3];
```

【パラメータ】

info[out] 初期化失敗時の詳細情報格納エリアの先頭アドレスを指定します。

【機能説明】

nx_init()は、NXACPの処理立ち上げおよび共通テーブルの初期化処理をします。ユーザは、システム立ち上げ時、最初にこのマクロを起動する必要があります。

【戻り値】

nx_init()は正常終了時、値0を返します。

異常終了の場合は、次の値を返します。なお、詳細情報格納エリアに設定される値との対応は、「付録A リターンコード詳細」を参照してください。

0x002 : NXACPは初期化済みです。

0x003 : 構築情報がロードされていません。

0x004 : NXACPシステムタスクがロードされていません（起動失敗含む）。

0x1e1 : ネットワークモジュールタイプまたはIPアドレスが異常です。

0x1e3 : 二重化LAN構成時、ネットワークモジュールタイプが異常です。

0x1f1 : ソケットマクロで異常を検出しました。

0x2e1 : NXACP状態遷移中です。

【補足】

- nx_init()が発行された直後から、nx_get()は発行可能となります。
- NXACPでは、ネットワークモジュールの実装チェックをnx_init()で実施します。したがって、未実装のUNOを定義した場合、エラーとして処理を中断します。
- nx_init()では、二重化されたデータフィールドでも片系LANで異常を検出した場合は、初期化処理を中断します。

3.2 nx_dfup

【名前】

nx_dfup — データフィールドの立ち上げ

【形式】

```
#include <nxacp.h>
long nx_dfup (dfn, nmode, mmode, info)
```

```
long *dfn;
long *nmode;
long *mmode;
long info[3];
```

【パラメータ】

dfn[in] 初期化対象のデータフィールド番号を指定します。
nmode[in] ノードモードを指定します。
mmode[in] メッセージモードを指定します。
info[out] 初期化失敗時の詳細情報格納エリアの先頭アドレスを指定します。

【機能説明】

nx_dfup()は、指定された1つのデータフィールドをOPEN状態にします。メッセージを送受信するには、nx_dfup()を発行して、使用したいデータフィールドをOPEN状態にしなければなりません。なお、生存信号の送出もnx_dfup()で開始します。

nmodeおよびmmodeに指定できる値とその意味を以下に示します。

- nmode
 - 0 : オンラインモード
 - 1 : テストモード
- mmode (nmodeがテストモード (=1) の場合だけ有効)
 - 0 : オンラインメッセージだけ
 - 1 : テストメッセージだけ

nx_dfup()は、データフィールド1つに対して有効ですので、使用するデータフィールド数分発行する必要があります。

nx_dfup()が異常終了した場合、指定したデータフィールドはCLOSEのままとなります。このとき、障害を検知するまで確保したポートなどのリソースは、すべて解放して処理を終了します。

構築情報誤りによる異常の場合は、構築情報を見直し修正してください。特にポート番号は他サブシステムと競合しないよう設定してください。

また、二重化LAN構成時、二重化された両方のLANが正常に初期化できない場合、異常として処理を終了します。片系のネットワークだけでOPEN状態にはしません。速やかに障害を取り除き、再試行してください。

【戻り値】

`nx_dfup()`は正常終了時、値0を返します。

異常終了の場合は、次の値を返します。なお、詳細情報格納エリアに設定される値との対応は「付録A リターンコード詳細」を参照してください。

0x001：NXACPが初期化されていません。

0x003：構築情報がロードされていません。

0x004：NXACPシステムタスクがロードされていません（タスク起動失敗含む）。

0x101：`dfn`の値が指定可能範囲外です。

0x102：`dfn`のデータフィールドが定義されていません。

0x104：`dfn`のデータフィールドは初期化済みです。

0x106：ソースデータフィールドが定義されていません。

0x107：ソースデータフィールドが初期化されていません。

0x161：`nmode`の指定誤りです。

0x162：`mmode`の指定誤りです。

(`nmode=0`：オンライン時は「0」を指定してください。0以外を指定しても無効です)

0x1e6：構築情報で指定した対応UNOのモジュールはすべて障害状態です。

0x1f1：ソケットマクロで異常を検知しました。

0x2e1：NXACPの状態遷移中です。

0x2f1：`dfn`のデータフィールドは状態遷移中です。

0x2f2：ソースデータフィールドは状態遷移中です。

【補足】

`nx_dfup()`が発行された直後から、ネットワークからの受信メッセージの取り込みが開始されます。

この後、長時間`nx_get()`が開始されないと、バッファオーバーフローが発生する可能性が高くなります（メッセージ受信ユーザタスクがIDLE/DORMANT中の場合は、NXACPで受信メッセージを廃棄します）。

また、EASへのサブルーチン登録は、`nx_dfup()`発行前に登録してください。`nx_dfup()`処理中に異常を検知した場合には、異常終了としてエラーリターンしますが、その際、ネットワーク関連の設定は、`nx_dfup()`が発行される前の状態まで戻した状態で制御をユーザに戻します。内蔵Etherでの障害発生についてはEASに障害通知されます。どの部位で障害が発生したかについては、`nx_dfup()`のエラーリターンコードだけでなくエラーログ情報からも解析できるようにすることを推奨します。

`nx_dfdwn()`を発行しないまま、同じデータフィールドに対して`nx_dfup()`を発行するとエラーリターンします（0x104でエラーリターン）。このとき、2回目に発行した`nx_dfup()`の引数で指定されたモード情報は設定されず、1回目に発行した引数のモード情報が有効のままとなります。

自ノード内通信を行いたい場合には、自ノード内専用のデータフィールドであるデータフィールド番号0のデータフィールドをOPENする必要があります。

【補足】の続き

- `nx_dfup()`では、二重化されたデータフィールドでも片系LANで異常を検出した場合は、初期化処理を中断します。
- マルチキャストグループ定義ファイル (`mgnN`) で、ポートを未使用 (=0) 指定として、このマクロを発行した場合、`0x1f1`のエラーコードで異常終了します。

3. 3 nx_dfdwn

【名前】

nx_dfdwn — データフィールドのCLOSE

【形式】

```
#include <nxacp.h>
long nx_dfdwn (dfn)

long *dfn;
```

【パラメータ】

dfn[in] 停止対象のデータフィールド番号を指定します。

【機能説明】

nx_dfdwn()は、指定された1つのデータフィールドをOPEN状態からCLOSE状態にします。ただし、ローカルデータフィールドを指定した場合、該当データフィールドに接続するリモートデータフィールドもCLOSE状態にします。

nx_dfdwn()によって、ユーザタスクからのメッセージの送信要求の受け付け、および該データフィールド宛メッセージの受信処理を停止し、処理待ちメッセージ（送信待ち、受信待ち、組み立て中）をパージします。この場合、nx_dfdwn()対象のデータフィールドに対し、nx_get()発行中のタスクへは、リターンコード0x202で通知されます。したがって、再度nx_get()を発行することによって、他のデータフィールドからメッセージを受信することができます。

shutdown予告生存信号を送出し、他ノードに状態変化を通知します。

このnx_dfdwn()は、データフィールド1つに対して有効ですので、OPENしているデータフィールドの数分発行してください。

【戻り値】

nx_dfdwn()は正常終了時、値0を返します。

異常終了の場合は、次の値を返します。

0x001 : NXACPが初期化されていない、または停止中です。

0x003 : 構築情報がロードされていません。

0x101 : dfnの値が指定可能範囲（0～255）外です。

0x102 : dfnのデータフィールドが定義されていません。

0x103 : dfnのデータフィールドが初期化されていません。

0x2e1 : NXACPが状態遷移中です。

0x2f1 : dfnのデータフィールドが状態遷移中です。

0x2f2 : ソースデータフィールドが状態遷移中です。

0x2f3 : リモートデータフィールドが状態遷移中です。

【補足】

nx_dfdwn()処理中、ポートなどのCLOSE処理などに異常を検知した場合でも異常終了しません。

nx_dfdwn()では、SHUTDOWNモードの生存信号を一定時間に3回送信しますので、終了するまで、5秒～10秒程度かかります。

3.4 nx_quit

【名前】

nx_quit — NXACPの停止

【形式】

```
#include <nxacp.h>
```

```
long nx_quit()
```

【パラメータ】

なし

【機能説明】

nx_quit()は、NXACPを停止状態にします。

nx_dfdown()を未発行で起動した場合、全データフィールドを停止します。

nx_dfdown()を起動しないままnx_quit()を発行した場合、ユーザタスクからのメッセージの送信要求の受け付けおよび全データフィールド宛メッセージの受信処理を停止し、SHUTDOWNモードの生存信号を送信して他ノードに状態変化を通知します。

【戻り値】

nx_quit()は正常終了時、値0を返します。

異常終了の場合は、次の値を返します。

0x001 : NXACPが初期化されていません。

0x003 : 構築情報がロードされていません。

0x2e1 : NXACPが状態遷移中です。

0x2f1 : 各データフィールドのどれかが遷移処理中です。

遷移中は、以下のイベント処理中を意味します。

nx_dfup()処理中、nx_dfdown()処理中。

【補足】

nx_quit()処理中、ポートなどのCLOSE処理などに異常を検知した場合でも異常終了しません。

nx_quit()もnx_dfdown()同様、SHUTDOWNモードの生存信号を一定時間に3回送信しますので、終了するまで、5秒～10秒程度かかります。

第4章 共有メモリマクロ

4.1 nx_init_tm

【名前】

nx_init_tm — 転写メモリの初期化

【形式】

```
#include <nxacp.h>
```

```
long nx_init_tm( df, tmid, tmmmap)
```

```
long *df;
```

```
long *tmid;
```

```
struct tmmmap *tmmmap;
```

tmmmap構造体の詳細は以下のとおりです。

```
struct tmmmap {  
    long m_type;  
    long m_mcg;  
    long m_cflags;  
    long m_memid;  
    long m_mincycm;  
    long m_caseno;  
    long m_maxblk ;  
    struct {  
        long m_cycm;  
        long m_blkno;  
        long m_blkcnt;  
    } m_send[m_caseno];  
} tmmmap;
```

【パラメータ】

df : データフィールド番号を1~最大データフィールドの範囲で指定します。

自ノード内通信用のDF=0またはリモートデータフィールドでは使用できません。

tmid : 転写メモリの識別子を1~4の範囲で指定します。

m_type : メモリ転写種別を指定します。

TMEM_SOFT(0) : 周期送信型ソフト転写

TMEM_FAST(2) : 分割送信型ソフト転写

m_mcg : マルチキャストグループ番号を1~最大マルチキャストグループ番号の範囲で指定します。

m_cflags : 転写メモリ制御フラグを指定します。下記オプションを使用する場合に設定してください。使用しない場合は「0」を指定してください。

CMF_ACL(1) : 排他制御ありの指定です。

(高速ソフト転写では指定できません。)

CMF_COLERR(2) : 書き込みエリアの重複チェックありの指定です。

制御を両方使用する場合は、「or」で指定してください

(CMF_ACL | CMF_COLERR)。

m_memid : ソフト転写の場合、転写メモリに割り付けるCMまたはGLBのアドレスを指定します。

m_mincyctm : サイクリック通信の最小(下記m_cyctmで最小の値)周期単位を指定します。この値は、同じTMIDを使用する全計算機で同じ値を指定します。

ソフト転写時は無効ですので「0」を指定してください。

m_caseno : 下記m_send[N]有効送信エリア数(1~9)を指定してください。

このCPUから転写メモリを送信しない場合は「0」を指定します。この場合、m_send[]の設定は不要です。

m_maxblk : 転写メモリで使用する最大ブロック番号を指定します。

m_send[m_caseno] : 転写メモリの送信エリア情報を指定します。

```
struct {
```

```
    long m_cyctm; 転写エリアごとの転写周期を指定します。
```

```
                ソフト転写時秒単位(ミリ秒の単位で指定してください)。
```

```
                例: 1秒=1000)。
```

```
    long m_blkno; 転写メモリ先頭からの先頭ブロック番号を指定します。
```

```
                ソフト転写時: 0~16383
```

```
    long m_blkcnt; 転写メモリ内のブロック数を指定します。
```

```
                ソフト転写時は1~255の範囲で指定します。
```

```
};m_send[m_caseno];
```

【機能説明】

転写メモリの属性を指定し、転写メモリの物理メモリと対応付けます。
システム立ち上げ時または新規転写メモリ作成時に発行します。

【留意事項】

転写時の転写ケース数、転写サイズ、転写周期、および接続ノード数は、CPUの負荷を上げる要因になりますので、システム設計時適切な値を設定してください。

【戻り値】

`nx_init_tm()`は正常終了時、値0を返します。
異常終了の場合は、次の値を返します。

0x001 : NXACPが停止中です。

0x003 : 制御テーブルがローディングされていません。

0x101 : データフィールド番号が範囲外です。

0x102 : データフィールド番号が定義されていません。

0x103 : データフィールドが初期化されていません。

0x105 : データフィールドのタイプがリモートデータフィールドです。

0x111 : マルチキャストグループ番号が範囲外です。

0x112 : マルチキャストグループが定義されていません。

0x501 : TMIDが範囲外です。

0x502 : マクロ発行順に誤りがあります。

0x503 : 転写メモリの種類に誤りがあります。

0x504 : 転写メモリ制御フラグの指定に誤りがあります。

0x505 : 転写メモリの周期に誤りがあります。

0x506 : 送信エリア数が範囲外です。

0x507 : ブロック番号が範囲外です。

0x508 : ブロック数が範囲外です。

0x509 : 指定アドレスが2バイト境界ではありません。

0x50A : 自ノード内の送信エリアが重複しています。

0x511 : 送信エリアの定義が転写メモリの範囲外です (`m_blkno`と`m_blkcnt`が不整合です)。

0x513 : 転写メモリがグローバルエリア外です。

【補足】

以下にソフト転写パラメータの違いを示します。

表3-2 パラメータ一覧

転写種別	周期送信型ソフト転写	分割送信型ソフト転写
パラメータ	設定可能範囲	
df	1~255	
tmid	1~4	
m_type	TMEM_SOFT	TMEM_FAST
m_mcg	1~255	
m_cflags	1~3	0、2
m_memid	GLBアドレス	
m_mincycle	—	
m_caseno	0~9	
m_maxblk	1~16384	
m_cyctm	ミリ秒 (*)	
m_blkno	0~16383	
m_blkcnt	1~255	

—の部分は無効です。
0を設定してください。

(*) 周期送信型ソフト転写は、100ms~86400000msです。

分割送信型ソフト転写は、10ms~86400000msです。

4. 2 nx_ctl_tm

【名前】

nx_ctl_tm — 転写メモリの制御

【形式】

```
#include <nxacp.h>
long nx_ctl_tm(df, tmid, cmd, hrtn)
```

```
long *df, *tmid, *cmd, *hrtn;
```

【パラメータ】

df : データフィールド番号

tmid : 転写メモリの識別子

cmd : メモリ転写種別

TM_START(2) : 指定TMIDの送受信処理開始

TM_READ(1) : 指定TMIDの受信処理だけ開始

TM_STOP(0) : 指定TMIDの転写停止

hrtn : ハード転写処理時の異常要因コード格納エリア (ハード転写未サポートのため値は格納されません)

【機能説明】

転写メモリの制御機能として次の機能を持ちます。

開始/停止前にnx_init_tmマクロで該転写メモリを初期設定しなければ発行できません。

- 転写開始
送受信両方の転写を開始します。
- 受信開始
受信処理だけの転写を開始します。
- 転写停止
転写を停止します。

【戻り値】

nx_ctl_tm()は正常終了時、値0を返します。

異常終了の場合は、次の値を返します。

0x001 : NXACPが停止中です。

0x003 : 制御テーブルがローディングされていません。

0x101 : データフィールド番号が範囲外です。

0x102 : データフィールド番号が定義されていません。

0x105 : データフィールドのタイプがリモートデータフィールドです。

0x501 : TMIDが範囲外です。

0x502 : マクロ発行順に誤りがあります。

0x504 : 転写メモリ制御フラグの指定に誤りがあります。

4. 3 nx_get_tm

【名前】

nx_get_tm ー 転写メモリ情報の取り出し

【形式】

```
#include <nxacp.h>
long nx_get_tm(df, tmid, tmmmap)
```

```
long *df, *tmid;
struct tmmmap *tmmmap;
```

【パラメータ】

df : 取り込みたいデータフィールド番号 (1~最大DF)
tmid : 取り込みたい転写メモリの識別子 (1~最大TMID)
tmmmap : 取り込み情報格納エリアのアドレス

【機能説明】

転写メモリの属性を指定エリアに取り込みます。
tmmmap構造体は「4. 1 nx_init_tm」を参照してください。

【戻り値】

nx_get_tm()は正常終了時、tmmmapに取り込んだ値を格納し、値0を返します。
異常終了の場合は、次の値を返します。

- 0x001 : NXACPが停止中です。
- 0x003 : 制御テーブルがローディングされていません。

- 0x101 : データフィールド番号が範囲外です。
- 0x102 : データフィールド番号が定義されていません。
- 0x105 : データフィールドのタイプがリモートです。

- 0x501 : TMIDが範囲外です。
- 0x502 : 転写メモリIDが初期化されていません。

4. 4 nx_write_tm

【名前】

nx_write_tm — 転写メモリへの書き込み

【形式】

```
#include <nxacp.h>  
long nx_write_tm(df, tmid, offset, da, dc, hrtn)
```

```
long *df;  
long *tmid;  
long *offset;  
long *da;  
long *dc;  
long *hrtn;
```

【パラメータ】

df : 書き込みする転写メモリの属すデータフィールド番号

tmid : 書き込みする転写メモリの識別子 (TMID番号)

offset : 転写メモリ内の相対バイト数 (2バイト境界でなければなりません)

da : ユーザデータエリアのアドレス (2バイト境界でなければなりません)

dc : ユーザデータエリアのサイズ

hrtn : ハード転写処理時の異常要因コード格納エリア (ハード転写未サポートのため値は格納されません)

【機能説明】

dfとtmidで指定された転写エリアに、daとdcで指定されたデータを書き込みます。

ソフト転写において排他制御を指定した場合、16320バイトまでデータの整合性を保証します。

ソフト転写で排他制御を使用しない場合、または高速ソフト転写のときはoffset、da、dc共に4の倍数の場合だけ4バイトの保証となります。

【戻り値】

`nx_write_tm()`は正常終了時、値0を返します。

異常終了の場合は、次の値を返します。

0x001 : NXACPが停止中です。

0x003 : 制御テーブルがローディングされていません。

0x101 : データフィールド番号が範囲外です。

0x102 : データフィールド番号が定義されていません。

0x105 : データフィールドのタイプがリモートです。

0x141 : オフセットサイズの指定に誤りがあります。

0x501 : TMIDが範囲外です。

0x502 : マクロ発行順に誤りがあります。

0x50B : daの指定に誤りがあります。

0x50C : dcの指定に誤りがあります。

0x531 : 転写メモリの確保に失敗しました。

4. 5 nx_read_tm

【名前】

nx_read_tm — 転写メモリの読み込み

【形式】

```
#include <nxacp.h>
```

```
long nx_read_tm(df, tmid, offset, da, dc, hrtn)
```

```
long *df;
```

```
long *tmid;
```

```
long *offset;
```

```
long *da;
```

```
long *dc;
```

```
long *hrtn;
```

【パラメータ】

df : 読み込みする転写メモリの属すデータフィールド番号

tmid : 読み込みする転写メモリの識別子 (TMID番号)

offset : 読み込む転写メモリ内の相対バイト数 (2バイト境界でなければなりません)

da : 読み込むユーザデータエリアのアドレス (2バイト境界でなければなりません)

dc : 読み込むユーザデータエリアのサイズ

hrtn : ハード転写処理時の異常要因コード格納エリア (ハード転写未サポートのため値は格納されません)

【機能説明】

dfとtmidで指定された転写エリアから、daとdcで指定されたデータを読み出します。

このとき、ソフト転写において排他制御を指定した場合、16320バイトまでデータの整合性を保証します。ソフト転写で排他制御を指定しない場合、または高速ソフト転写のときはoffset、dadc共に4の倍数の場合だけ4バイトの保証となります。

【戻り値】

`nx_read_tm()`は正常終了時、値0を返します。

異常終了の場合は、次の値を返します。

0x001 : NXACPが停止中です。

0x003 : 制御テーブルがローディングされていません。

0x101 : データフィールド番号が範囲外です。

0x102 : データフィールド番号が定義されていません。

0x105 : データフィールドのタイプがリモートです。

0x141 : オフセットサイズの指定に誤りがあります。

0x501 : TMIDが範囲外です。

0x502 : マクロ発行順に誤りがあります。

0x50B : daの指定に誤りがあります。

0x50C : dcの指定に誤りがあります。

0x512 : 指定DCが転写メモリ範囲外です。

0x531 : 転写メモリの確保に失敗しました。

このページは白紙です。

付録

付録A リターンコード詳細

(1) マルチキャスト通信マクロリターンコード

(1/2)

コード	内容	ユーザ対処方法
0x0001	NXACPが初期化されていない、または停止中	NXACPを初期化してください。
0x0002	NXACP初期化済み	NXACPはすでに初期化済みです（初期化不要）。
0x0003	制御テーブルがローディングされていない	構築および構築情報をローディングしてください。
0x0004	NXACPタスクがローディングされていない、または起動失敗	NXACP本体を再ローディングしてください。 (*1)
0x0101	指定データフィールド範囲外	データフィールド番号を確認後、再試行してください。
0x0102	指定データフィールドが定義されていない	データフィールド番号を確認後、再試行してください。
0x0103	指定データフィールドが初期化されていない	データフィールド番号を確認後、再試行してください。
0x0104	指定データフィールド初期化済み	データフィールド番号を確認後、再試行してください。
0x0105	指定データフィールドタイプ誤り	データフィールド番号を確認後、再試行してください。
0x0106	ソースデータフィールドが定義されていない	ソースデータフィールド番号を確認後、再試行してください。
0x0107	ソースデータフィールドが初期化されていない	ソースデータフィールド番号を確認後、再試行してください。
0x0108	PU間通信の送信先PUのDFが初期化されていない	送信先PUのDFを初期化した後、再試行してください。
0x0111	指定マルチキャストグループ番号範囲外	マルチキャストグループ番号を確認後、再試行してください。
0x0112	指定マルチキャストグループが定義されていない	マルチキャストグループ番号を確認後、再試行してください。
0x0121	指定トランザクションコード番号範囲外	トランザクションコード番号を確認後、再試行してください。
0x0131	指定メッセージ/バッファサイズ範囲外	メッセージ/バッファサイズを確認後、再試行してください。
0x0141	指定オフセットサイズ範囲外	オフセットサイズを確認後、再試行してください。
0x0151	指定タイムアウト監視時間範囲外	タイムアウト時間を確認後、再試行してください。
0x0161	指定ノードモード誤り	ノードモードを確認後、再試行してください。
0x0162	指定メッセージモード誤り	メッセージモードを確認後、再試行してください。
0x01E1	デバイス種別またはIPアドレス異常	ユニット番号、IPアドレスを確認後、再試行してください。
0x01E3	二重化時のネットワーク組み合わせ異常	UNOに対応するデバイス確認後、再試行してください。
0x01E6	全モジュールが障害状態	モジュールを交換後、再試行してください。
0x01F1	ソケットマクロ異常	総合編マニュアルを参照してください。 (*2)

(2/2)

コード	内容	ユーザ対処方法
0x0201	送信バッファ空なし	リトライしてください。
0x0202	受信処理中でデータフィールドDOWN	リトライしてください。
0x0211	送信不可状態	モジュールを交換してください。
0x0231	PU間通信で異常を検出	リトライしてください。
0x02E1	NXACP状態遷移中	リトライしてください。
0x02F1	指定データフィールド状態遷移中	リトライしてください。
0x02F2	ソースデータフィールド状態遷移中	リトライしてください。
0x02F3	リモートデータフィールド状態遷移中	リトライしてください。
0x0301	マクロ発行タスク番号範囲外	マクロを発行できるタスク番号は1~208までです。
0x0302	受信TCDが定義されていない	該当タスクの受信TCDを定義してください。
0x0303	送信TCDが定義されていない	送信権チェックTCD定義を確認後、再試行してください。

(*1) 0x0004には、タスクの起動失敗も含まれています。タスクの起動とは、QUEUEとTIMERを指します。

TIMERに関しては、NXACPでは2タスクをタイマ登録します。したがって、ユーザがNXACP起動前にシステム限界数までタイマにタスクを登録している場合もこのエラーで処理を中断します。

(*2) 詳細コード格納エリアにRCTLNETが返すリターンコードが格納されています。このリターンコードを「ユーザーズマニュアル S10VE総合編 (マニュアル番号 SEJ-1-001)」で確認してください。

付録A リターンコード詳細

<リターンコードとマクロの対応>

○ : エラーリターンあり - : エラーリターンなし

コード	nx_init	nx_dfup	nx_dfdwn	nx_quit	nx_put	nx_get		INFO1	INFO2	INFO3
0x0001	-	○	○	○	○	○		-	-	-
0x0002	○	-	-	-	-	-		-	-	-
0x0003	○	○	○	○	○	○		-	-	-
0x0004	○	○	-	-	-	-		TN	-	-
0x0101	-	○	○	-	-	-		-	-	-
0x0102	-	○	○	-	○	-		-	-	-
0x0103	-	-	○	-	○	-		-	-	-
0x0104	-	○	-	-	-	-		-	-	-
0x0105	-	-	-	-	-	-		-	-	-
0x0106	-	○	-	-	-	-		-	-	-
0x0107	-	○	-	-	-	-		-	-	-
0x0108	-	-	-	-	○	-		-	-	-
0x0111	-	-	-	-	○	-		-	-	-
0x0112	-	-	-	-	○	-		-	-	-
0x0121	-	-	-	-	○	-		-	-	-
0x0131	-	-	-	-	○	○		-	-	-
0x0141	-	-	-	-	-	○		-	-	-
0x0151	-	-	-	-	-	○		-	-	-
0x0161	-	○	-	-	-	-		-	-	-
0x0162	-	○	-	-	-	-		-	-	-
0x01E1	○	-	-	-	-	-		UNO	-	-
0x01E3	○	-	-	-	-	-		DFNO	-	-
0x01E6	-	○	-	-	-	-		UNO1	UNO2	-
0x01F1	○	○	-	-	-	-		UNO	MCODE	ERRNO
0x0201	-	-	-	-	○	-		-	-	-
0x0202	-	-	-	-	-	○		-	-	-
0x0211	-	-	-	-	○	-		-	-	-
0x0231	-	-	-	-	○	-		-	-	-
0x02E1	○	○	○	○	-	-		-	-	-
0x02F1	-	○	○	○	-	-		-	-	-
0x02F2	-	○	○	-	-	-		-	-	-
0x02F3	-	-	○	-	-	-		-	-	-
0x0301	-	-	-	-	○	○		-	-	-
0x0302	-	-	-	-	-	○		-	-	-
0x0303	-	-	-	-	○	-		-	-	-

INFO1~3は ()、nx_init()/nx_dfup()のパラメータで、エラーリターンする場合の詳細情報です。

TN : タスク番号

UNO : ユニット番号 (コード=0x1E1の場合、上位2バイトと下位2バイトの2LANのUNOを各々格納する)

MCODE : RCTLNETのマクロコード

SOCKET: 1

BIND: 2

GETSOCKOPT: 3

SETSOCKOPT: 4

ERRNO : RCTLNETマクロのリターンコード (*)

SLOTNO : スロット番号

DFNO : データフィールド番号

(*) マクロのリターンコードについては、「ユーザズマニュアル S10VE総合編 (マニュアル番号 SEJ-1-001)」を参照してください。

【エラーコード体系】

マルチキャスト通信マクロのエラーコード体系を以下に示します。なお、×は任意の数字です。

0x00××：未インストールまたはインストール失敗に伴うエラーです。

0x01××：パラメータチェックによるエラーです。

ただし、イニシャライズ要求（`nx_init()`、`nx_dfup()`）ではハードウェアエラーを含みません。

0x02××：タイミングエラーです。リトライすることによって、救済される可能性があるエラーです。

ただし、リトライするタイミングには注意してください。

即リトライして、同じタイミングエラーになった場合、無限ループとなる可能性があります。

0x03××：ユーザのシステム設計誤りによるエラーです。

(2) 共有メモリマクロリターンコード

コード	内容	ユーザ対処方法
0x0501	TMID範囲外	TMIDを確認後、再試行してください。
0x0502	マクロ発行手順誤り	各マクロの発行手順を確認後、再試行してください。
0x0503	転写種類指定誤り	転写種類を確認後、再試行してください。
0x0504	転写制御フラグ指定誤り	転写制御フラグを確認後、再試行してください。
0x0505	転写周期指定誤り	転写周期を確認後、再試行してください。
0x0506	送信エリア数範囲外	送信エリア数を確認後、再試行してください。
0x0507	ブロック番号範囲外	ブロック番号を確認後、再試行してください。
0x0508	ブロック数範囲外	ブロック番号を確認後、再試行してください。
0x0509	指定アドレス指定誤り (奇数アドレス)	指定アドレスを確認後、再試行してください。
0x050A	自ノード内送信エリア重複	送信エリアを確認後、再試行してください。
0x050B	DA指定誤り	DAを確認後、再試行してください。
0x050C	DC指定誤り	DCを確認後、再試行してください。
0x0531	転写メモリ確保失敗 (rserv失敗)	転写エリアへのアクセスタスク数を15以内にしてください。
0x0511	送信エリアの定義が転写メモリの範囲外	blknoとblkcntを確認後、再試行してください。
0x0512	指定DCが転写メモリの範囲外	DCを確認後、再試行してください。
0x0513	転写メモリがグローバルエリアの範囲外	指定アドレスを確認後、再試行してください。

【エラーコード体系】

共有メモリマクロのエラーコード体系を以下に示します。なお、×は任意の数字です。

0x050×：パラメータチェックによるエラーです。

0x053×：システム動作に異常がある場合のエラーです。

<リターンコードとマクロの対応>

○：エラーリターンあり　－：エラーリターンなし

コード	nx_init_tm	nx_ctl_tm	nx_get_tm	nx_write_tm	nx_read_tm
0x0001	○	○	○	○	○
0x0003	○	○	○	○	○
0x0101	○	○	○	○	○
0x0102	○	○	○	○	○
0x0103	○	－	－	－	－
0x0105	○	○	○	○	○
0x0111	○	－	－	－	－
0x0112	○	－	－	－	－
0x0141	－	－	－	○	○
0x0501	○	○	○	○	○
0x0502	○	○	○	○	○
0x0503	○	－	－	－	－
0x0504	○	○	－	－	－
0x0505	○	－	－	－	－
0x0506	○	－	－	－	－
0x0507	○	－	－	－	－
0x0508	○	－	－	－	－
0x0509	○	－	－	－	－
0x050A	○	－	－	－	－
0x050B	－	－	－	○	○
0x050C	－	－	－	○	○
0x0511	○	－	－	－	－
0x0512	－	－	－	－	○
0x0513	○	－	－	－	－
0x0531	－	－	－	○	○

(3) 構築コマンド (confnxsv) エラーコード一覧

- confnxsvコマンドで表示するメッセージフォーマットは以下のとおりです。

Error/Warning : 障害メッセージ 障害No.

- 詳細を以下の表に示します。

Errorとなったまま、tblldnxsvコマンドを実行した場合、tblldnxsvも異常終了します。

障害No.	障害メッセージ	障害内容
1300	AAA is out of range (file:BBB, line:CCC)	AAAが範囲外です。 (BBB : ファイル名, CCC : 行No.)
1302	AAA is multiple defined (file:BBB)	AAAが重複しています。 (BBB : ファイル名)
1303	AAA is undefined (file:BBB)	AAAが未定義です。(BBB : ファイル名)
1330	Illegal format of name (file:BBB, line:CCC) "AAA"	"AAA" という名称はフォーマットに合いません。 (BBB : ファイル名, CCC : 行No.)
1331	Illegal format of numeric value (file:BBB, line:CCC)	設定値がフォーマットに合いません。 (BBB : ファイル名, CCC : 行No.)
1332	Not enough entries (file:BBB, line:CCC)	登録値が足りません。 (BBB : ファイル名, CCC : 行No.)
1333	Max over entries (file:BBB) "AAA"	"AAA" が最大値を超えています。 (BBB : ファイル名)
1356	Resource data unmatched (file:BBB) "AAA"	"AAA" に食い違いがあります。 (BBB : ファイル名)
1651	Specified AAA is already defined (file:BBB, line:CCC)	AAAはすでに定義されています。 (BBB : ファイル名, CCC : 行No.)
2653	Specified file is for AAA (file:BBB)	指定したファイルはAAA用です。 (BBB : ファイル名)
9001	Specified size is not on AAA byte boundary (file:BBB, line:CCC)	設定値はAAAの倍数になっていません。 (BBB : ファイル名, CCC : 行No.)
9002	Can't open file "BBB"	BBBファイルがOPENできません。

CP側構築情報の各エラーに対しては、障害内容を確認後、再試行してください。

HP側構築情報の各エラーに対しては、HP側のnxacp.confの定義情報「MAXDFCNT」に「0」を設定し、再試行してください。

付録B ログフォーマット

EASに通知するエラー情報フォーマットを以下に示します。

(1) 障害通知一覧

イベント概要	コード	障害通知内容	ユーザ対処方法
バッファステータス報告	0x0401	送受信バッファの使用率がしきい値に到達または回復しました。	バッファ面数を増加してください。
プロトコル異常	0x0201	受信したメッセージのNXヘッダに異常を検知し、該当メッセージを廃棄しました。	送信元ノードと構築情報を確認してください。
ソケット障害	0x0501	ソケットマクロから異常が報告されました。	システムマネージャに報告してください。
転写エリア重複異常	0x0601	転写の送信エリアの重複を検知しました（ソフト転写時だけ）。	転写の送信アドレスを見直してください。

【補足】

プロトコル異常（0x0201）を通知するチェック内容は以下のとおりです。

チェック内容	廃棄条件	通知有無	備考
受信サイズ	1473バイト以上（最大）	あり	(*1)
受信メッセージ	パターン（NUXM）異常	あり	
送信先アドレス	DF不一致	あり	DAをチェック
	MGN不一致	あり	DAをチェック
送信元アドレス	送信元ノード未定義	なし	(*2)
受信MCG	MGN未定義	なし	
TCD	TCD未定義	なし	
MODE	ポートモード不一致	あり	

(*1) 受信したNXヘッダが63バイト以下のときには、通知は行いません（NXACPで廃棄するだけです）。

(*2) ノード未定義（範囲外）についても通知はしません。該当ノードからのメッセージ受信不要と判定します。

(2) EASの入力データフォーマット (ADB)

詳細は「S10VE ソフトウェアマニュアル CPMS概説&マクロ仕様 (マニュアル番号 SEJ-3-201)」を参照してください。

	名称	説明
0	logno	エラーログ番号
4	timestamp	時刻
8	type	重要度タイプ (エラータイプによって異なります)
12	class	障害検出コンポーネントクラス (NXACPは0x2011)
16	retcode	無効 (NXACPは0固定)
20	errtype	故障種別 (NXACPは10固定)
22	flag	エラーメッセージフラグ (NXACPは0固定)
24	site[16]	サイト名称
40	erb[117] (*)	エラーブロック (障害報告データ)。 エリアサイズは468バイト固定ですが、有効データ サイズはエラータイプによって異なります。
512	dhpbuf[128]	DHPデータ (512バイト)
1024		

(*) erbは、先頭から4バイトごとにformtype、size、errorcode、詳細データ1、データ2...という構成になります。

(3) 障害詳細データフォーマット一覧

名称	バッファステータス報告	プロトコル異常	ソケット障害	転写エラー重複異常
erb内formtype	0x0103	0x0102	0x0104	0x0105
erb内size	64	76	16	24
erb内errorcode	0x0401	0x0201	0x0501	0x0601
詳細データ (各情報は4 バイト単位)	DFN	DFN	DFN	DFN
	SPEAK	PORTNO	DADDR	TMID
	RPEAK	NXHD (*)	DPORT	CASENO
	SOVFCNT			BLKNO
	SHWCCNT			BLKCNT
	SLWCCNT			
	ROVFCNT			
	RHWCCNT			
	RLWCCNT			
	SOVFCNT			
	SHWTCNT			
	SLWTCNT			
	ROVTCNT			
	RHWTCNT			
RLWTCNT				
type	NOTE	WARNING	NONFATAL	NONFATAL

(*) NXHDは64バイト構成で、NXヘッダを指します。構成は「付録F メッセージヘッダフォーマット」を参照してください。

DFN：データフィールド番号

SPEAK：送信バッファピーク数

RPEAK：受信バッファピーク数

SOVFCNT：送信バッファオーバフローカウンタ

SHWCCNT：送信バッファHW到達カウンタ

SLWCCNT：送信バッファLW回復カウンタ

ROVFCNT：受信バッファオーバフローカウンタ

RHWCCNT：受信バッファHW到達カウンタ

RLWCCNT：受信バッファLW回復カウンタ

SOVTCNT：送信バッファオーバフロートータルカウンタ

SHWTCNT：送信バッファHW到達トータルカウンタ

SLWTCNT：送信バッファLW到達トータルカウンタ

ROVTCNT：受信バッファオーバフロートータルカウンタ

RHWTCNT：受信バッファHW到達トータルカウンタ

RLWTCNT：受信バッファLW回復トータルカウンタ

PORTNO：ポート番号

DADDR：宛先アドレス

DPORT：宛先ポート

TMID：転写メモリ識別子

CASENO：ケース番号

BLKNO：ブロック番号

BLKCNT：ブロック数

MERRNO：マクロエラーコード

(4) 表示フォーマット

- バッファステータス報告

```
%NX-I-SOFT-0103 SITE=xxxxxxxxxxxxx RC=00000000 yyyy/mm/dd hh:mm:ss LOG=xxx  
EC=000000401 Buffer status  
DFN=XXXXXXXXX SPEAK=XXXXXXXXX RPEAK=XXXXXXXXX  
SOVFCNT=XXXXXXXXX SHWCCNT=XXXXXXXXX SLWCCNT=XXXXXXXXX  
ROVFCNT=XXXXXXXXX RHWCCNT=XXXXXXXXX RLWCCNT=XXXXXXXXX  
SOVFCNT=XXXXXXXXX SHWTCNT=XXXXXXXXX SLWTCNT=XXXXXXXXX  
ROVFCNT=XXXXXXXXX RHWTCNT=XXXXXXXXX RLWTCNT=XXXXXXXXX  
~
```

RC: Return Code

EC: Error Code

DFN: Data Field Number

SPEAK: Send Buffer Peak Count

RPEAK: Receive Buffer Peak Count

SOVFCNT: Send Buffer Overflow Current Count

SHWCCNT: Send Buffer High-Water Current Count

SLWCCNT: Send Buffer Low-Water Current Count

ROVFCNT: Receive Buffer Overflow Current Count

RHWCCNT: Receive Buffer High-Water Current Count

RLWCCNT: Receive Buffer Low-Water Current Count

SOVFCNT: Send Buffer Overflow Total Count

SHWTCNT: Send Buffer High-Water Total Count

SLWTCNT: Send Buffer Low-Water Total Count

ROVFCNT: Receive Buffer Overflow Total Count

RHWTCNT: Receive Buffer High-Water Total Count

RLWTCNT: Receive Buffer Low-Water Total Count

- ・プロトコル異常

```
%NX-W-SOFT-0102 SITE=xxxxxxxxxxxxxx RC=00000000 yyyy/mm/dd hh:mm:ss LOG=xxx
EC=000000201 Message frame error
DFN   =XXXXXXXXX PORTNO=XXXXXXXXX
H_TYPE =XXXXXXXXX ML =XXXXXXXXX SA   =XXXXXXXXX DA =XXXXXXXXX
V_SEQ  =XXXXXXXXX SEQ =XXXXXXXXX M_CTL=XXXXXXXXX
INQ_ID1 =XXXXXXXXX INQ_ID2=XXXXXXXXX INQ_ID3=XXXXXXXXX
TCD    =XXXXXXXXX G_TID1 =XXXXXXXXX G_TID2 =XXXXXXXXX
MSGMD  =XXXXXXXXX BLOCK =XXXXXXXXX FU1   =XXXXXXXXX
~
```

EC: Error Code

DFN: Data Field Number

PORTNO: Port Number

以下の情報については、NXヘッダを示します。

構成は「付録F メッセージヘッダフォーマット」を参照してください。

H_TYPE: HEAD Pattern

ML: Message Length

SA: Source Address

DA: Destination Address

V_SEQ: Version Sequence number

SEQ: Send Sequence number

M_CTL: Message Control

INQ_ID1: Inquiry Identifier1

INQ_ID2: Inquiry Identifier2

INQ_ID3: Inquiry Identifier3

TCD: Transaction Code

G_TID1: Global Transaction Identifier1

G_TID2: Global Transaction Identifier2

MSGMD: Message Mode

BLOCK: Message Block

FU1: Future Use

・ソケット障害

```
%NX-E-SOFT-0104 SITE=xxxxxxxxxxxxxx RC=00000000 yyyy/mm/dd hh:mm:ss LOG=xxx  
EC=000000501 Socket error  
DFN =XXXXXXXX DADDR=XXXXXXXX DPORT=XXXXXXXX  
~
```

RC: Return Code

EC: Error Code

DFN: Data Field Number

DADDR: Destination IP Address

DPORT: Destination Port Number

・転写エリア重複異常

```
%NX-E-SOFT-0105 SITE=xxxxxxxxxxxxxx RC=00000000 yyyy/mm/dd hh:mm:ss LOG=xxx  
EC=000000601 Transfer memory address error  
DFN =XXXXXXXX TMID =XXXXXXXX CASENO =XXXXXXXX  
BLKNO =XXXXXXXX BLKCNT =XXXXXXXX  
~
```

RC: Return Code

EC: Error Code

DFN: Data Field Number

TMID: Transfer Memory Identifier

CASENO: Send Case Number

BLKNO: Send Block Number

BLKCNT: Send Block Count

付録C ノード状態変化通知フォーマット

他ノードの状態変化通知は、NXACPがリザーブしている通知専用IRSUBにリンクします。

状態変化通知のリンクするIRSUB番号は332番です。ユーザは332番にサブルーチンを登録しておくことでノードの状態変化をリアルタイムで検知できます。なお、このサブルーチンはNXACPの起動前に登録してください。

(1) 通知フォーマット

	名称	説明
0	CODE	イベント番号
4	DFN	データフィールド番号
8	LNN	論理ノード番号
12	LANNO	LAN番号
16	TYPE	変化要因タイプ
20		

(2) 詳細情報

生通知：初めて生存信号を受信したか、死と認識した後に生存信号を受信したことを意味します。

死通知：一度生通知を受信してから規定時間内に一度も生存信号を受信しなかったことを意味します。

死予告通知：SHUTDOWN予告またはメンテナンス予告の生存信号を受信したことを意味します。

	生通知	死通知	死予告通知
0	0x0301	0x0302	0x0302
4	DFN	DFN	DFN
8	LNN	LNN	LNN
12	LANNO (*)	LANNO (*)	LANNO (*)
16	0固定	2固定	1固定
20			

(*) 1LAN構成では1固定です。自ノードについても1固定です。

2LAN構成では、UNO1のLANの場合=1、UNO2のLANの場合=2を設定します。
自ノードについては1固定で、一度だけ報告します。

【注意】

このIRSUBは、NXACPシステムタスクのサブルーチンとして動作します。

したがって、このIRSUBの中で待ちになるマクロを発行しないでください。また、使用可能なスタックサイズは512バイトまでです。

付録D DHPレコード一覧

取得ポイント	コード	データ数	データ1	データ2	データ3	データ4
nx_init()開始	0x00400001	0	—	—	—	—
nx_init()終了	0x00400021	1	RTNCD	—	—	—
nx_dfup()開始	0x00400002	3	DFN	NMODE	MMODE	—
nx_dfup()終了	0x00400022	2	DFN	RTNCD	—	—
nx_dfdown()開始	0x00400003	1	DFN	—	—	—
nx_dfdown()終了	0x00400023	1	RTNCD	—	—	—
nx_quit()開始	0x00400004	0	—	—	—	—
nx_quit()終了	0x00400024	1	RTNCD	—	—	—
nx_put()開始	0x00400005	4	DFN	MGN	TCD	LEN
nx_put()終了	0x00400025	1	RTNCD	—	—	—
nx_get()開始	0x00400006	3	TMOUT	BFSZ	OFSET	—
nx_get()終了 (*)	0x00400026	4	RTNCD	SA	DA	TCD/LEN
受信後データ組み立て完了	0x00100009	2	POSTA	0固定	—	—
受信後データ組み立て未完	0x00100009	2	0固定	0固定	—	—
転写データ受信	0x00100009	2	0固定	1固定	—	—
生存信号受信	0x00100009	2	0固定	2固定	—	—
ヘッダ異常による廃棄	0x00100009	2	0固定	3固定	—	—

各データの意味は以下のとおりです。

RTNCD：処理結果

DFN：データフィールド番号

NMODE：ノードモード

MMODE：メッセージモード

MGN：マルチキャストグループ番号

TCD：トランザクションコード

LEN：メッセージサイズ

TMOUT：タイムアウト監視時間

BFSZ：バッファサイズ

OFSET：オフセット

SA：送信元アドレス

DA：送信先アドレス

POSTA：POSTアドレス

(*) nx_get終了で、メッセージなしまたはエラー終了時は、データ1から4の各データはすべて0（ゼロ）となります。

付録E 制御トレースについて

NXACPでは、RAS機能としてNXACP内部の動作をロギングする機能があります。

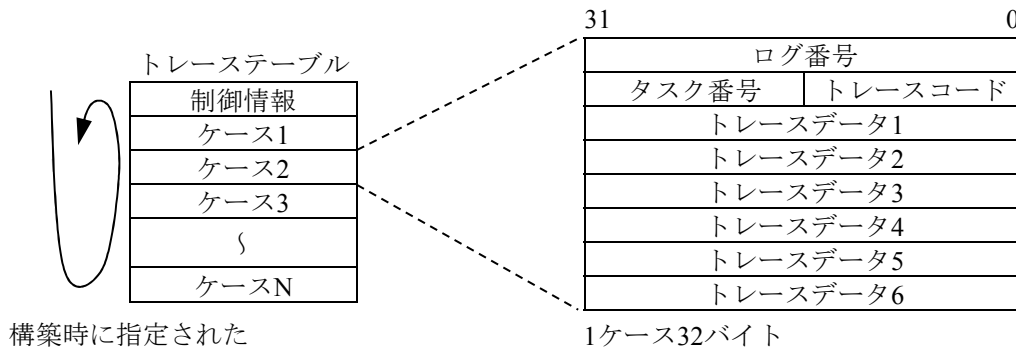
これを制御トレースと呼び、NXACP関連のトラブル発生時の解析データとして利用できます。

制御トレースを使用するか否かはユーザ指定ですが、テスト中はトレース取得し、オンライン時はトレースを停止させ、高速運転することも可能ですので、512ケース（16KB）程度確保することを推奨します。

トレースの表示はRPDPのダンプコマンド（sd）を使用します。

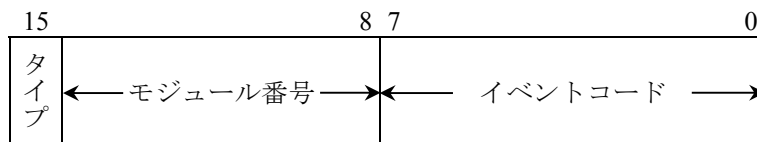
(1) トレースエリアフォーマット

トレースエリアのフォーマットは以下のとおりです。構築情報で指定されたケースを循環的に使用します。



構築時に指定された
ケース数を循環的に
使用します。

(2) トレースコードフォーマット

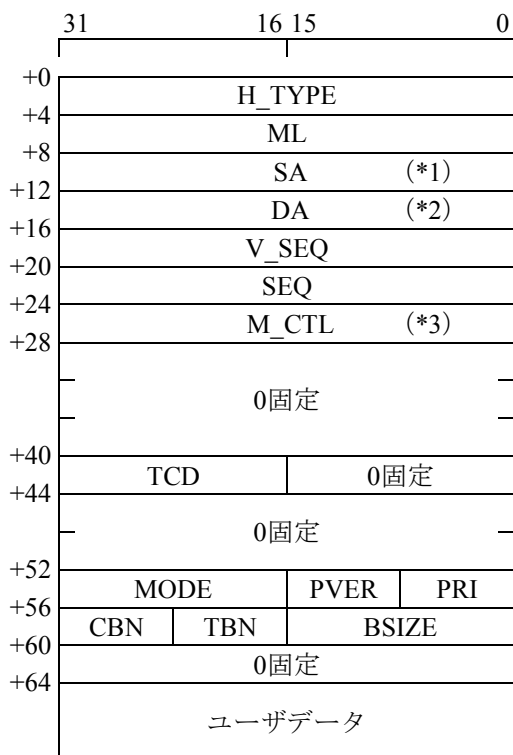


タイプ：エラートレース時は、このビットをONします。

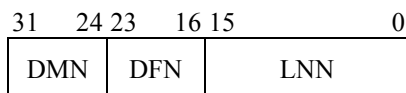
モジュール番号：以下のとおりです。

- | | | |
|------------------|-------------|---------------|
| 1 : nx_rcv | 16 : nx_put | 32 : nx_init |
| 2 : nx_upexe | 17 : nx_get | 33 : nx_quit |
| 3 : nx_htim | 18 : nx_abs | 34 : nx_dfup |
| 4 : nx_ltim | | 35 : nx_dfdwn |
| 5 : nx_dsnd | | 36 : nx_ins |
| 6 : nx_memac | | 37 : nx_cdon |
| 7 : nx_cycsnd | | 38 : nx_cdoft |
| 8 : nx_purcv | | 39 : nx_ctl |
| 10 : nx_init_tm | | |
| 11 : nx_ctl_tm | | |
| 12 : nx_get_tm | | |
| 13 : nx_write_tm | | |
| 14 : nx_read_tm | | |

付録F メッセージヘッダフォーマット



(*1) SA

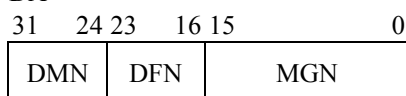


DMN：ドメイン番号（現状0固定）

DFN：データフィールド番号

LNN：送信元ノード番号

(*2) DA



DMN：ドメイン番号（現状0固定）

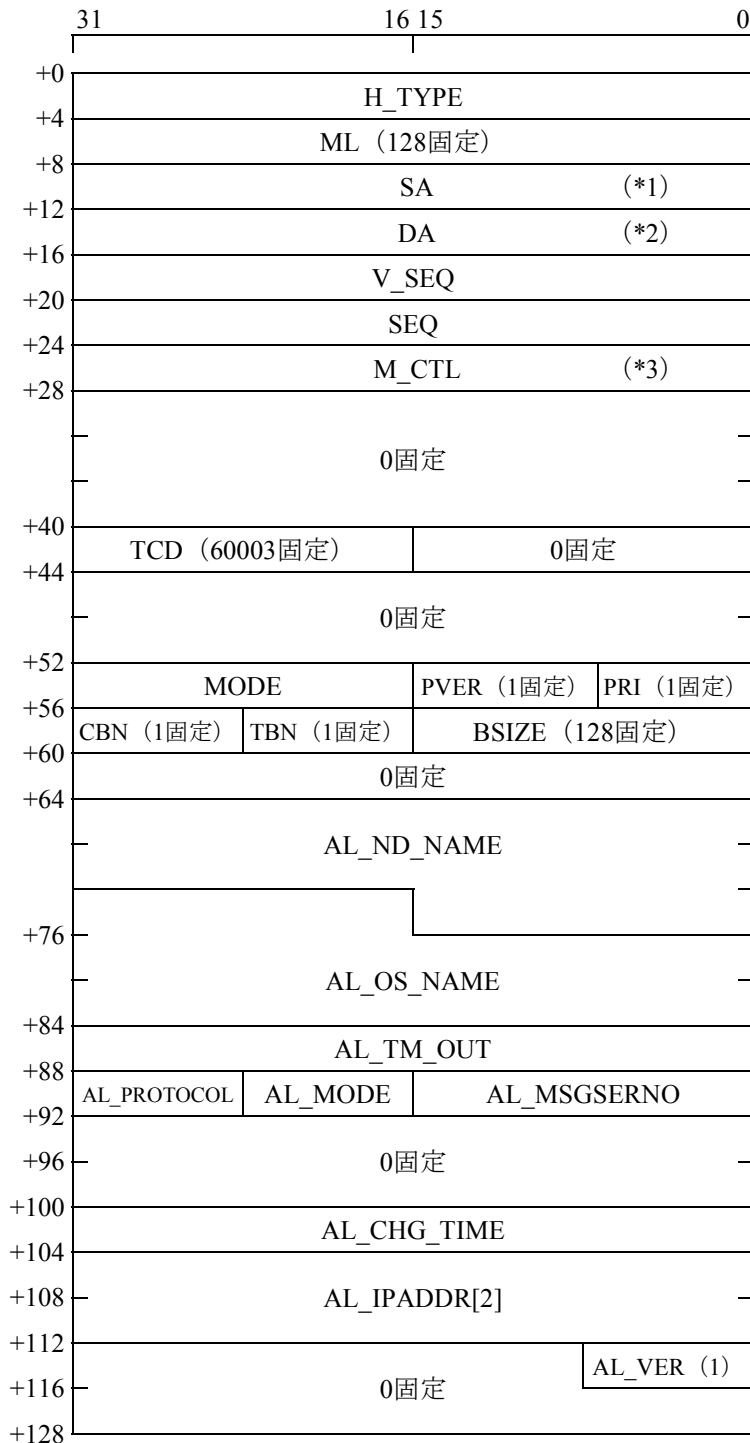
DFN：データフィールド番号

MGN：マルチキャストグループ番号

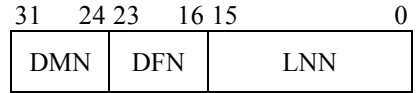
(*3) M_CTL：0x80000000固定（マルチキャスト通信）

記号名称	サイズ (バイト)	内容
H_TYPE	4	ヘッダタイプ（ASCIIで「NUXM」を設定）
ML	4	NXヘッダを含めたメッセージの長さ
SA	4	メッセージ送信元アドレス。（*1）参照。
DA	4	メッセージ送信先アドレス。（*2）参照。
V_SEQ	4	通番バージョン番号（通番が初期化された時刻を設定）
SEQ	4	メッセージ送信通番（値は0x00000001～0x7FFFFFFF）
M_CTL	4	メッセージ送信制御情報。（*3）参照。
TCD	2	トランザクションコード
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）
PVER	1	NXプロトコルバージョン（1固定）
PRI	1	メッセージの優先レベル（0固定）
CBN	1	カレントブロック番号
TBN	1	トータルブロック数
BSIZE	2	分割パケット内のユーザデータサイズ（フラグメントブロックのバイト数）

付録G 生存信号ヘッダフォーマット



(*1) SA

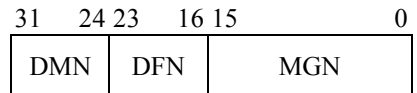


DMN : ドメイン番号 (現状0固定)

DFN : データフィールド番号

LNN : 送信元ノード番号

(*2) DA



DMN : ドメイン番号 (現状0固定)

DFN : データフィールド番号

MGN : マルチキャストグループ番号

(*3) M_CTL : 0x80000000固定

(マルチキャスト通信)

記号名称	サイズ (バイト)	内容
H_TYPE	4	ヘッダタイプ (ASCIIで「NUXM」を設定)
ML	4	NXヘッダを含めたメッセージの長さ (128固定)
SA	4	メッセージ送信元アドレス。 (*1) 参照。
DA	4	メッセージ送信先アドレス。 (*2) 参照。
V_SEQ	4	通番バージョン番号 (通番が初期化された時刻を設定)
SEQ	4	メッセージ送信通番 (値は0x00000001~0x7FFFFFFF)
M_CTL	4	メッセージ送信制御情報 (0x80000000固定)
TCD	2	トランザクションコード (60003固定)
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)
PVER	1	NXプロトコルバージョン (1固定)
PRI	1	メッセージの優先レベル (1固定)
CBN	1	カレントブロック番号 (1固定)
TBN	1	トータルブロック数 (1固定)
BSIZE	2	分割パケット内のユーザデータサイズ (128固定)
AL_ND_NAME	10	ノード名称 (最後にNULLが設定されたASCII文字列。最大9文字)
AL_OS_NAME	10	ベンダ機器名称 (HI_S10VE)
AL_TM_OUT	4	生存信号タイムアウト監視時間 (秒)
AL_MSGSERNO	2	生存報告メッセージ通番
AL_MODE	1	生存報告モード。 (*4) 参照。
AL_PROTOCOL	1	プロトコル種別 (4固定)
AL_CHG_TIME	4	ノード状態変化発生時間。 (*5) 参照。
AL_IPADDR	8	IPアドレス (AL_IPADDR[0]はLAN1、AL_IPADDR[1]はLAN2)
AL_VER	1	生存信号メッセージバージョン番号 (1固定)

(*4) AL_MODE=1 : 通常モード (生状態)

=2 : shutdown予告モード

(*5) AL_CHG_TIME : 立ち上げ時のグリニッジ時間で、年月日時分秒を設定します。

停止時は、「0」を設定します。