HITACHI
Inspire the Next

AIX

通信管理

XNF/AS　プログラマーズガイド NLI 編

解説・文法書

3000-3-B48-10

マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，巻末の「ソフトウェアマニュアルのサービス ご案内」をご参照ください。

■対象製品

適用 OS：AIX 5L
R-F1M141-518 XNF/AS/NLI 01-08
適用 OS：AIX
P-F1M14-51218 XNF/AS/NLI 02-01

■輸出時の注意

本製品を輸出される場合には，外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上，必要な手続きをお取りください。
なお，ご不明な場合は，弊社担当営業にお問い合わせください。

■商標類

AIX は，米国における International Business Machines Corporation の登録商標です。
AIX 5L は，米国およびその他の国における International Business Machines Corporation の商標です。
INS-C は，日本電信電話 (株) のサービス名称です。
INS-P は，日本電信電話 (株) のサービス名称です。
UNIX は，X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

■発行

2002 年 9 月 (第 1 版) 3000-3-B48
2009 年 12 月（第 2 版）3000-3-B48-10

■著作権

変更内容

変更内容（3000-3-B48-10） XNF/AS/NLI 01-08（**適用** OS：AIX 5L），XNF/AS/NLI 02-01（**適用** OS：AIX）

追加・変更内容	変更個所
64 ビットカーネルで動作できるようにしました。	―
RF パケット応答同期機能をサポートしました。	3.8.2，3.8.4，5.1，5.3.13，5.3.14，5.3.20，10.2，11.8

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは，通信管理プログラム XNF/AS/NLI の使用方法を説明したものです。
このマニュアルの対象となるプログラムプロダクトを次に示します。

適用 OS：AIX 5L
　R-F1M141-518　XNF/AS/NLI

適用 OS：AIX
　P-F1M14-51218　XNF/AS/NLI
このマニュアルでは，適用 OS が AIX 5L の XNF/AS/NLI を「XNF/AS/NLI V1 製品」，適用 OS が AIX の XNF/AS/NLI を「XNF/AS/NLI V2 製品」と記載します。

■対象読者
XNF/AS/NLI を使用して上位プログラムの設計や製造をする方を対象としています。
なお，UNIX の基礎的な知識，および OSI の第 3 層（ネットワーク層）に関する知識があることを前提としています。

■マニュアルの構成
このマニュアルは，次に示す章と付録から構成されています。

第 1 章　通信管理概説
通信管理の概要について説明しています。

第 2 章　ネットワークサポート
ネットワークの構成，サポート通信網について説明しています。

第 3 章　アプリケーションインタフェース
アプリケーションのインタフェースについて説明しています。

第 4 章　NLI のインタフェース
NLI のインタフェースについて説明しています。

第 5 章　ライブラリ関数
ライブラリ関数の文法について説明しています。

第 6 章　マクロ
マクロの文法について説明しています。

第 7 章　ネットワークアドレス仕様
ネットワークアドレスについて説明しています。

第 8 章　ネットワークプロトコルオプション
ネットワークプロトコルオプションについて説明しています。

第 9 章　エラーコード

提供関数が異常終了した場合のエラーコードについて説明しています。

第 10 章　プログラムの作成

プログラムの作成，およびインタフェース構造体のパラメタについて説明しています。

第 11 章　付加機能

XNF/AS/NLI の付加機能について説明しています。

付録 A　従来製品との差異

従来製品（HI-UX/WE2 上で動作する XNF/S-E2）との差異について説明しています。

■関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

- 通信管理　XNF/AS　解説・運用編（3000-3-B41）（AIX 5L 用）
- 通信管理　XNF/AS　構成定義編（3000-3-B42）（AIX 5L 用）
- 通信管理　XNF/AS　NSAP アドレス概説編（3000-3-B43）
- 通信管理　XNF/AS　解説・運用編（3000-3-B61）（AIX 用）
- 通信管理　XNF/AS　構成定義編（3000-3-B62）（AIX 用）
- AIX マニュアル（AIX に付属する CD-ROM マニュアルなど）

■読書手順

このマニュアルは，次の表に従ってお読み頂くことをお勧めします。

目的	記載個所
通信管理の概要について知りたい	1 章
ネットワークの構成，サポート通信網について知りたい	2 章
アプリケーションのインタフェースについて知りたい	3 章
XNF/AS/NLI が提供する関数，マクロ，またはインタフェースの構造体について知りたい	4 章～ 6 章
ネットワークアドレス，またはネットワークプロトコルオプションについて知りたい	7 章，8 章
エラーコードについて知りたい	9 章
プログラムの作成，およびインタフェース構造体のパラメタの設定・参照について知りたい	10 章
XNF/AS/NLI の付加機能について知りたい	11 章
従来製品との差異について知りたい	付録 A

■このマニュアルでの表記

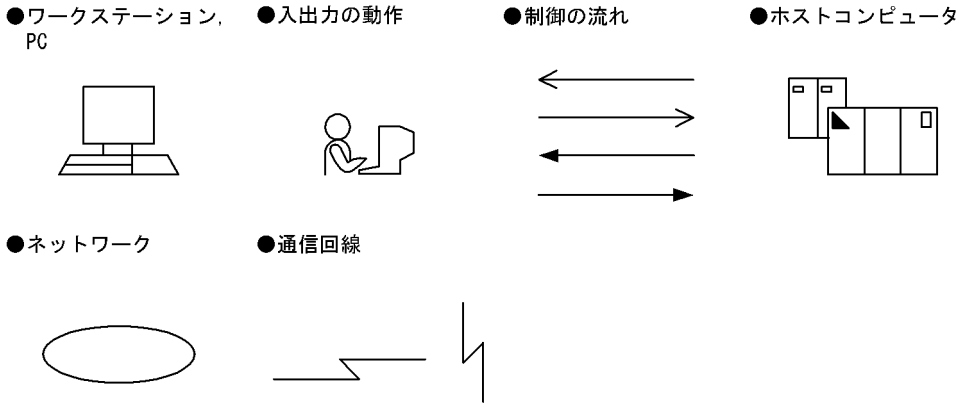
このマニュアルでは，製品名称を次に示す略称で表記しています。

製品名称		略称
AIX 5L V5.1	AIX 5L	AIX
AIX 5L V5.2		
AIX 5L V5.3		
AIX V6.1		
Extended HNA based Communication Networking Facility/for Advanced Server		XNF/AS
EXtended HNA based Communication Networking Facility/for Advanced Server/Network Layer Interface		XNF/AS/NLI または NLI

■図中で使用する記号

このマニュアルの図中で使用する記号を，次のように定義します。

原寸作図（Ｂ５判マニュアル用）図中で使用する記号の説明 (Ver.02-30)



■常用漢字以外の漢字の使用について

このマニュアルでは，常用漢字を使用することを基本としていますが，次に示す用語については，常用漢字以外の漢字を使用しています。

個所（かしょ） 必須（ひっす）

■ KB（キロバイト）などの単位表記について

1KB（キロバイト），1MB（メガバイト），1GB（ギガバイト），1TB（テラバイト）はそれぞれ 1,024 バイト，$1{,}024^2$ バイト，$1{,}024^3$ バイト，$1{,}024^4$ バイトです。

目次

図目次

表目次

1 通信管理概説

この章では，通信管理の概要について説明します。

1.1 NLI とは

NLI は，専用線，パケット網，ISDN 網などのネットワークを利用するユーザに，X.25 パケットプロトコルを直接利用するためのネットワーク層インタフェースを提供する通信管理プログラムです。

NLI を組み込むと，次のことができます。

（1） HNA/OSI に依存しない相手との通信

X.25 パケットプロトコルを直接アクセスできるため，HNA/OSI が掲載されていない相手との通信ができます。

（2） HNA/OSI との共存

X.25 の SPI・NSS を，他プロトコルと競合しない範囲で利用する場合には，同一回線上で HNA/OSI の共存ができます。

（3） 通信性能の向上

X.25 パケットプロトコルを直接アクセスできるため，HNA/OSI などの上位プロトコルのオーバヘッドがなくなり，スループットが向上します。

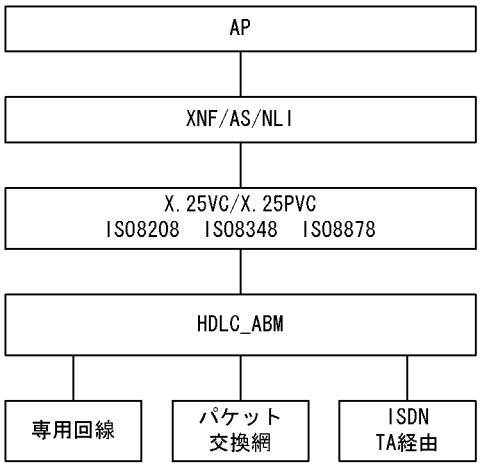
（4） 通信料金の削減

X.25 パケットプロトコルを直接アクセスできるため，HNA/OSI などの上位プロトコルのオーバヘッドがなくなり，パケット料金などが削減できます。

1.2 プロトコルとサービス

NLI がサポートするプロトコルとサービスを図 1-1 に示します。プロトコルとサービスは，ISO の規格番号で示してあります。

図 1-1 NLI がサポートするプロトコルとサービス



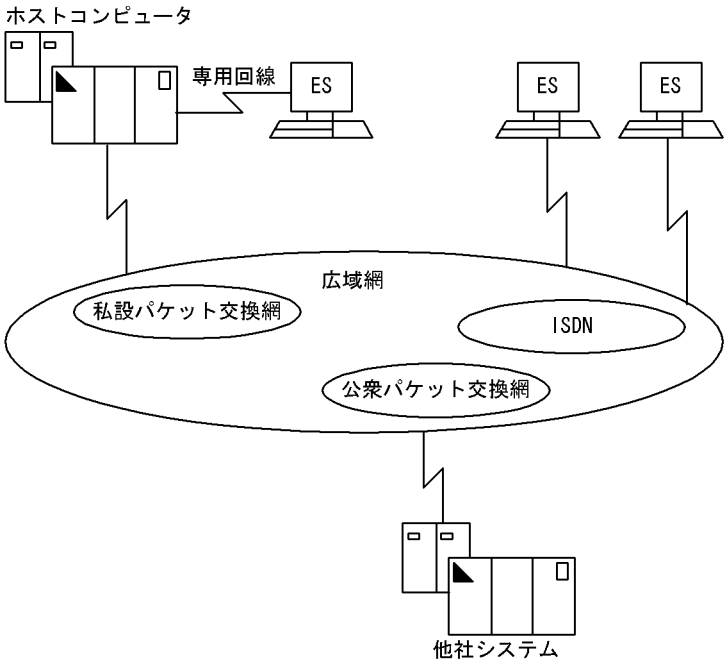
2 ネットワークサポート

この章では，ネットワークの構成，サポート通信網について説明します。

2.1 ネットワークの構成

ES（エンドシステム）とネットワークの接続構成を図 2-1 に示します。

図 2-1 ES とネットワークの接続構成



2.2 サポート通信網

NLI がサポートする通信網を表 2-1 に示します。

表 2-1 NLI がサポートする通信網

通信網種別	メディア / インタフェース
専用回線	X.25(84)VC
パケット交換網	X.25(80)PVC，X.25(80/84)VC
ISDN（INS ネット 64-TA 経由）	［パケット交換］ X.25(80)PVC，X.25(80/84)VC

3 アプリケーションインタフェース

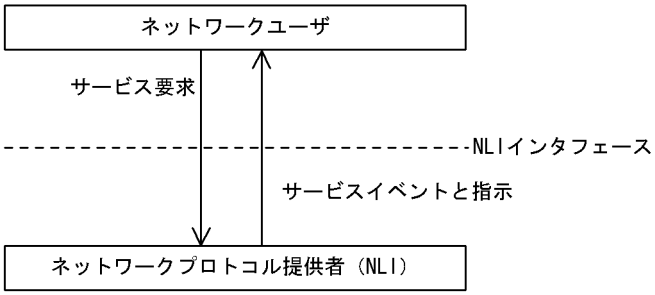
この章では，アプリケーションのインタフェースについて説明します。

3.1 NLI

NLI では，X.25 サービスインタフェースを直接使用できる NLI インタフェースをサポートしています。この NLI インタフェースは，C 言語のライブラリ関数として提供します。

二つのユーザプロセス間のデータ転送をサポートする NLI のサービスを図 3-1 に示します。

図 3-1　二つのユーザプロセス間のデータ転送をサポートする NLI のサービス



ネットワークプロトコル提供者（NLI）は，ネットワークインタフェースのサービスを提供し，ネットワークユーザはこれらのサービスを要求します。ネットワークユーザは，必要なサービス要求（データ転送要求など）を出すことによって，NLI のサービスをアクセスできます。また，NLI はデータの到着など，さまざまなイベントをユーザに通知します。

NLI は，ネットワークユーザとのインタフェースとして，次に示す手順を提供しています。

1. **ローカル管理手順**
2. **ネットワークコネクション確立手順**
3. **データ送受信手順**
4. **ネットワークコネクション解放手順**
5. **ネットワークコネクションリセット手順**
6. **優先データ受信手順**

3.2 ローカル管理

ローカル管理手順では，ネットワークユーザと NLI 間のローカルな手順を定義します。ネットワーク端点の制御および補助的なライブラリ関数を定義しています。

ネットワーク端点とは，ネットワークユーザがネットワークサービスを利用するときの NLI との結合点（通信チャネル）を示します。

NLI のローカル管理関数を表 3-1 に，ユーザとネットワークプロトコル提供者間のチャネルを図 3-2 に示します。また，ネットワークコネクションを確立する前までの手順を図 3-3 に，ネットワークコネクションを解放したあとの手順を図 3-4 に示します。

表 3-1 NLI のローカル管理関数

関数	説明
n_alloc	ネットワークインタフェースデータ構造体を割り当てます。
n_bind	ネットワーク端点にネットワークアドレスを結合します。
n_bind2	相手局ネットワークアドレスを指定して，ネットワークアドレスを結合します。
n_chglist	n_bind2 で指定した相手局ネットワークアドレスを変更します。
n_close	ネットワーク端点をクローズします。
n_error	ネットワークインタフェースエラーメッセージを表示します。
n_free	n_alloc で割り当てられた構造体を解放します。
n_getinfo	特定のネットワークプロバイダに関連する一連の情報を返します。
n_getstate	ネットワーク端点の状態を返します。
n_look	ネットワーク端点の現イベントを返します。
n_open	ネットワークプロバイダに接続されているネットワーク端点を確立します。
n_sync	ネットワークプロバイダとネットワーク端点を同期化します。
n_unbind	ネットワーク端点からネットワークアドレスの結合を解除します。

図 3-2 ユーザとネットワークプロトコル提供者間のチャネル

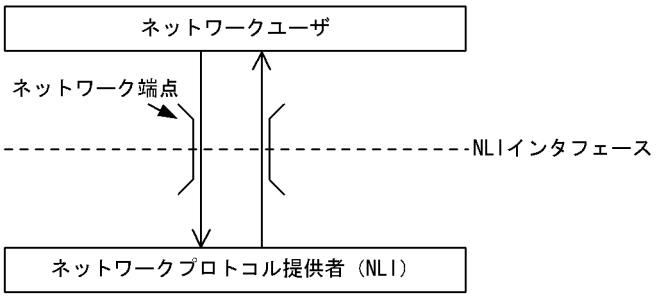


図 3-3　ネットワークコネクションを確立する前までの手順

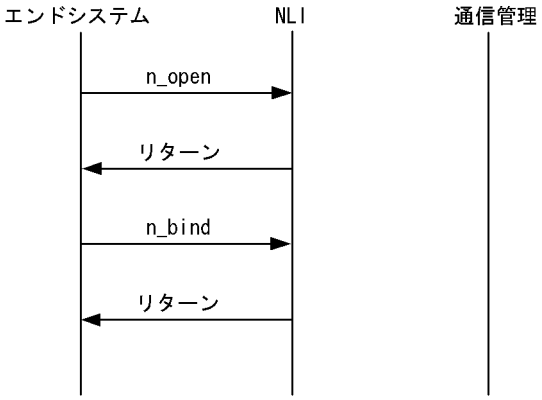
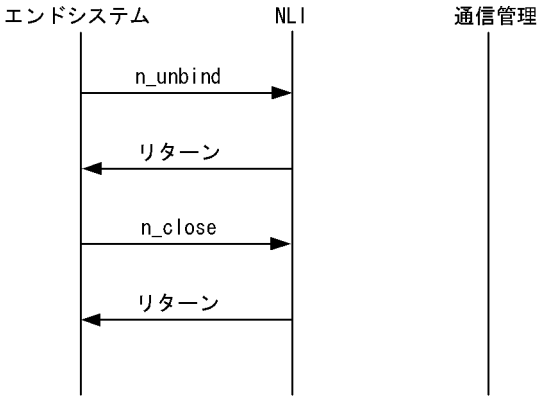


図 3-4　ネットワークコネクションを解放したあとの手順



3.3 ネットワークコネクションの確立

ネットワークコネクション確立フェーズでは，図 3-5 に示すように，二つのユーザ間のネットワークコネクションを確立します。n_connect 関数は，ユーザが接続したい相手のネットワークアドレスを指定することによって，ネットワークコネクションの確立要求を発行します。相手局は，n_listen 関数を発行して着信要求を受け取り，n_accept 関数で確立応答を返します。ネットワークコネクション確立フェーズで使用する関数を表 3-2 に示します。また，同期型の確立手順を図 3-6 に，非同期型の確立手順を図 3-7 に示します。

図 3-5 ネットワーク接続

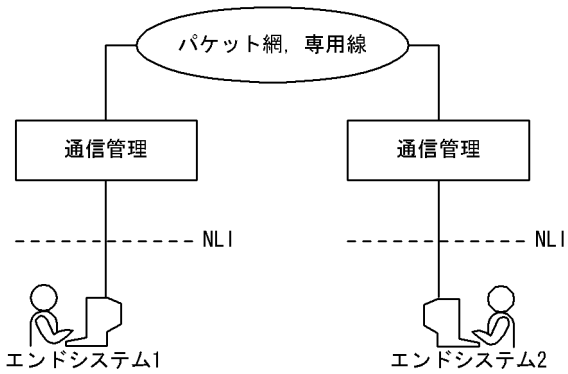


表 3-2 ネットワークコネクション確立フェーズで使用する関数

関数	説明
n_accept	ネットワーク接続に対する要求を受け入れます。
n_connect	指定されたあて先でネットワークユーザとの接続を確立します。
n_listen	ネットワークユーザからの接続要求の指示を待ちます。
n_rcvconnect	非同期モードで n_connect を発行したあとに，接続の確立を通知します。
select	相手局からのイベントを待ちます。詳細については，AIX マニュアルを参照してください。

図 3-6　同期型の確立手順

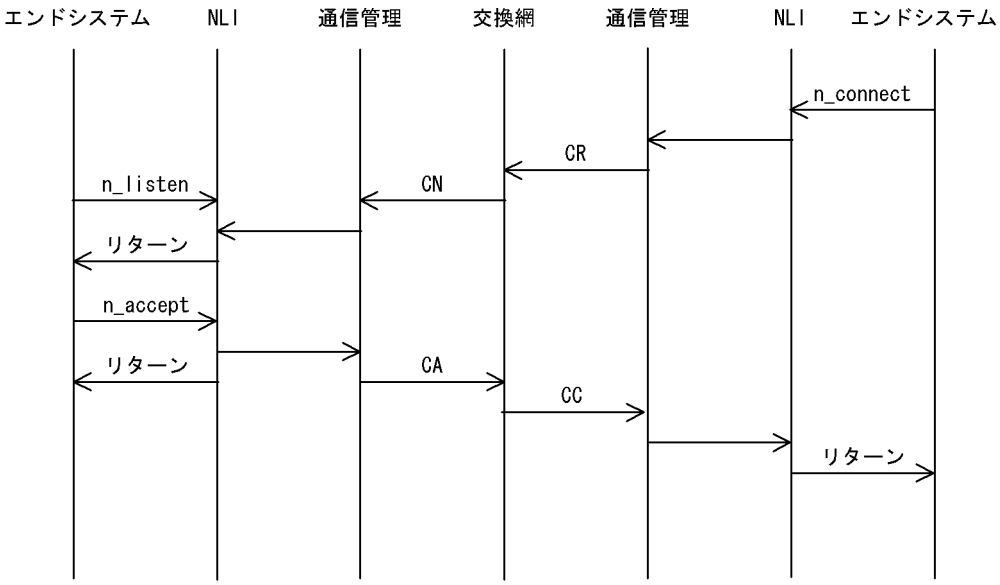
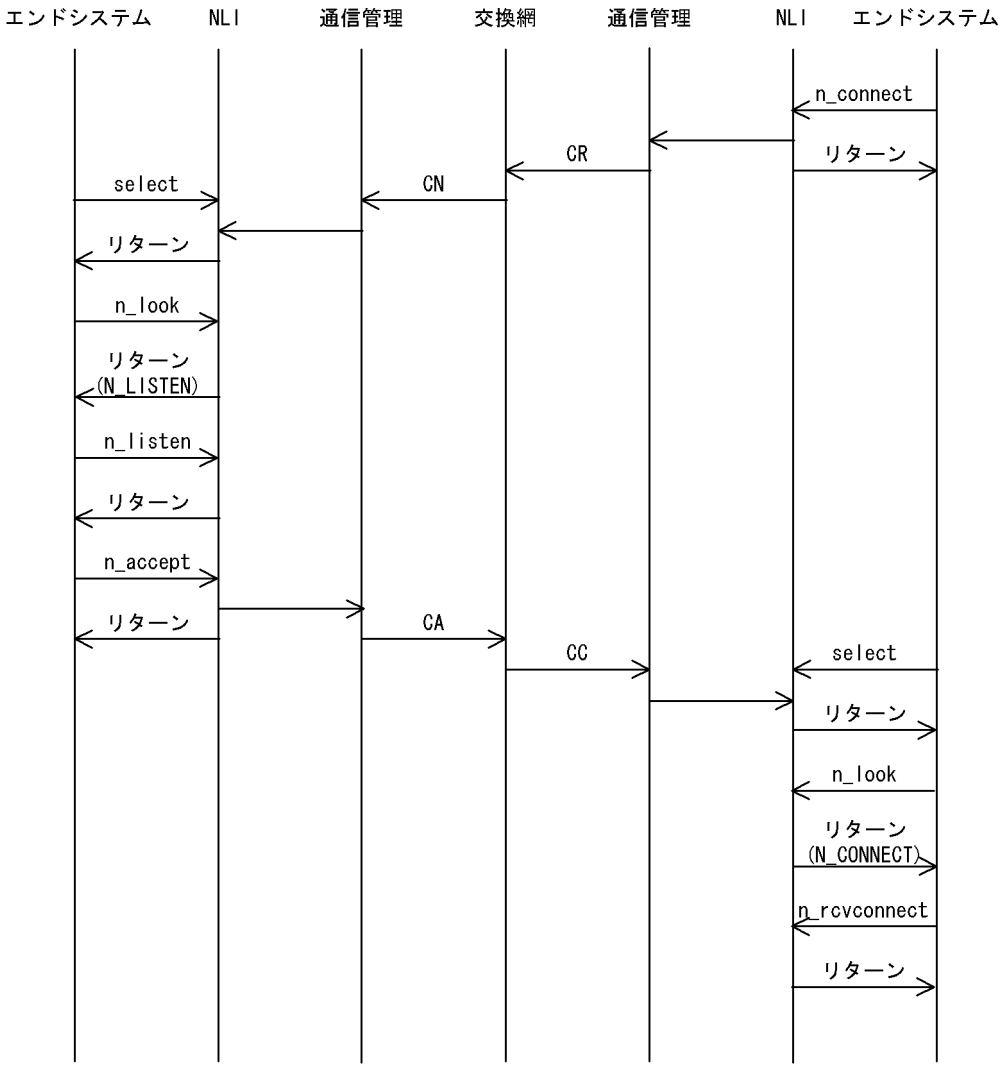


図 3-7　非同期型の確立手順



3.4 データの送受信

データの送受信フェーズでは，確立したネットワークコネクションを使用して，n_snd 関数，n_rcv 関数でユーザデータの送受信を行います。データ送受信で使用する関数を表 3-3 に示します。また，同期型のデータ送受信の手順を図 3-8 に，非同期型のデータ送受信の手順を図 3-9 に示します。

表 3-3 データ送受信で使用する関数

関数	説明
n_snd	ネットワークコネクションを介してデータを送信します。
n_rcv	ネットワークコネクションを介してデータを受信します。
select	相手局からのイベントを待ちます。送信ビジー状態が解除されるまで待ちます。詳細については，AIX マニュアルを参照してください。

図 3-8 同期型のデータ送受信の手順

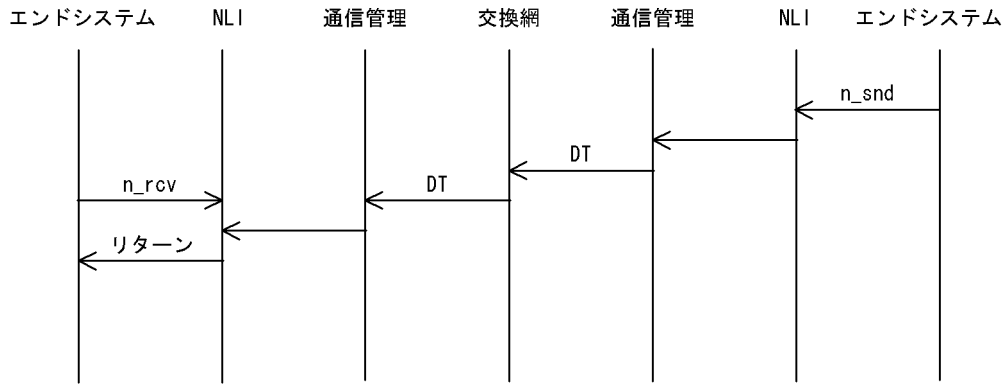
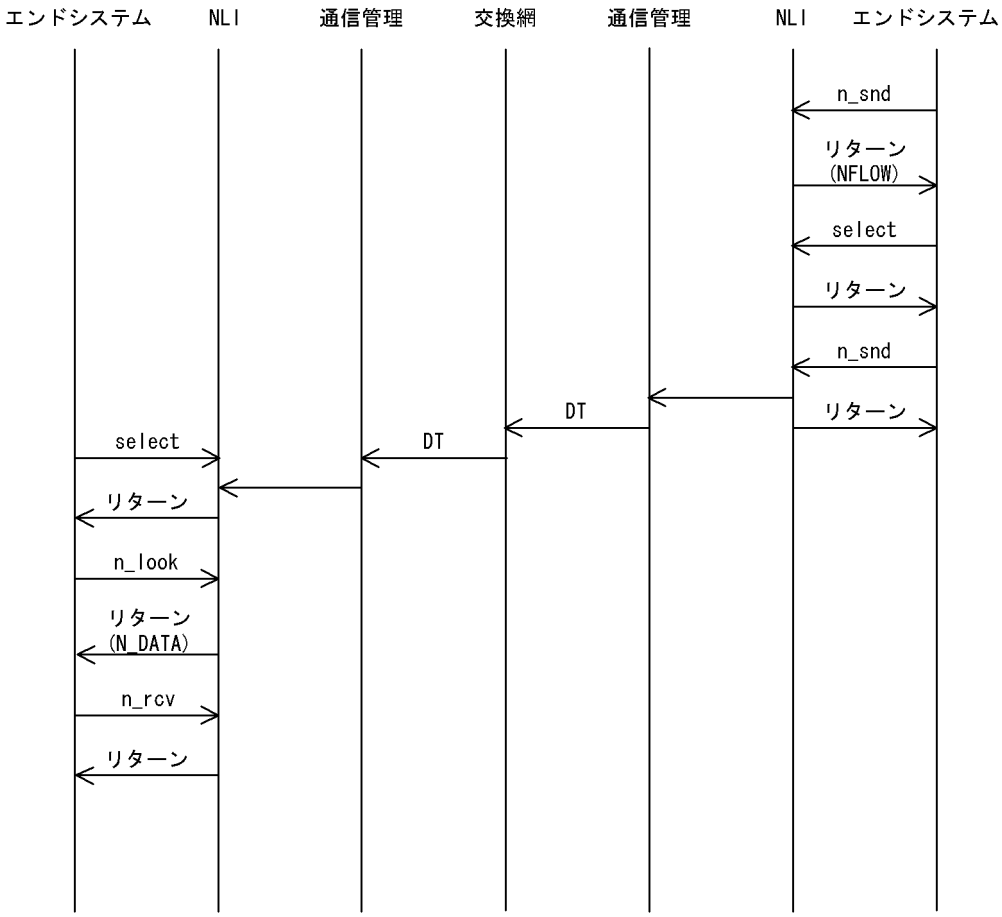


図 3-9　非同期型のデータ送受信の手順



3.5 ネットワークコネクションの解放

ネットワークコネクションの解放フェーズでは，確立しているネットワークコネクションを解放します。ネットワークコネクションの解放で使用する関数を表 3-4 に示します。また，同期型のネットワークコネクションの解放手順を図 3-10 に，非同期型のネットワークコネクションの解放手順を図 3-11 に示します。

表 3-4 ネットワークコネクションの解放で使用する関数

関数	説明
n_snddis	ネットワークコネクションの解放を要求します。
n_rcvdis	ネットワークコネクションの解放指示を受信します。

図 3-10 同期型のネットワークコネクションの解放手順

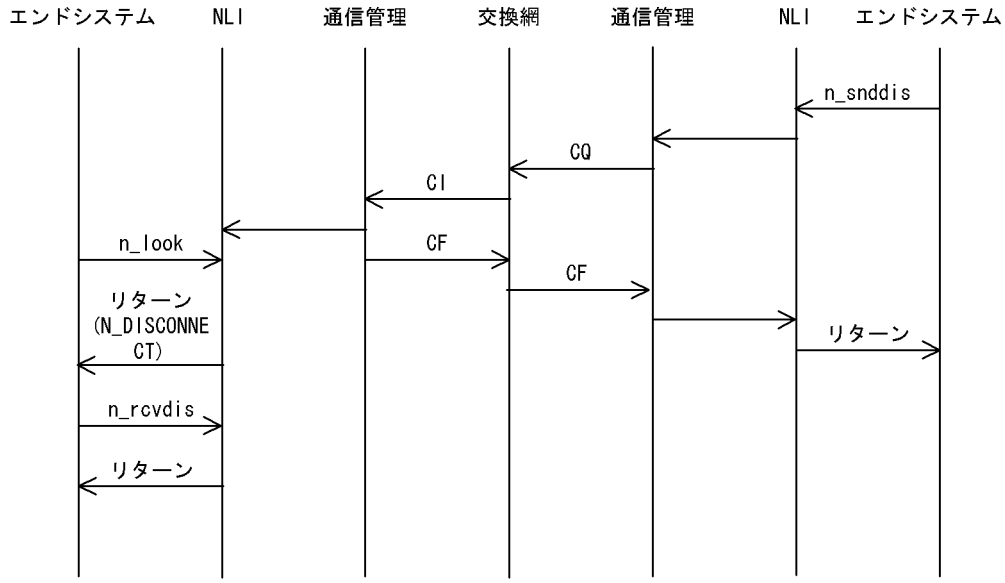
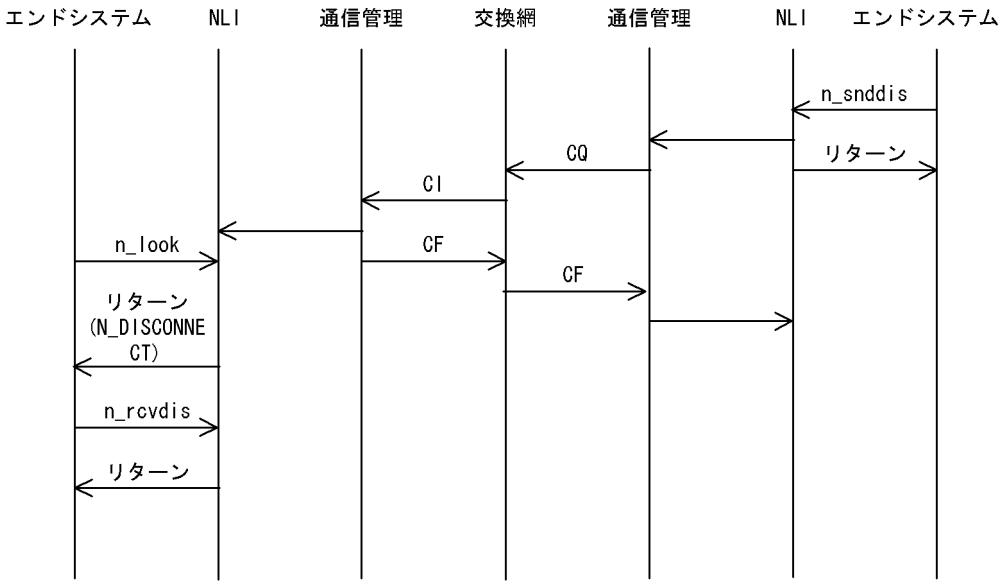


図 3-11　非同期型のネットワークコネクションの解放手順



3.6 ネットワークコネクションのリセット

ネットワークコネクションのリセットフェーズでは，確立しているネットワークコネクションをリセットします。ネットワークコネクションのリセットで使用する関数を表 3-5 に示します。また，同期型のネットワークコネクションのリセット手順を図 3-12 に，非同期型のネットワークコネクションのリセット手順を図 3-13 に示します。なお，VC 接続ではサポートしていません。

表 3-5 ネットワークコネクションのリセットで使用する関数

関数	説明
n_sndrst	ネットワークコネクションのリセットを要求します。
n_rcvrst	ネットワークコネクションのリセット確認を受信します。
n_look	ネットワークコネクションのリセット指示を受信して，リセット応答を発行します。
select	相手局からのイベントを待ちます。詳細については，AIX マニュアルを参照してください。

図 3-12 同期型のネットワークコネクションのリセット手順

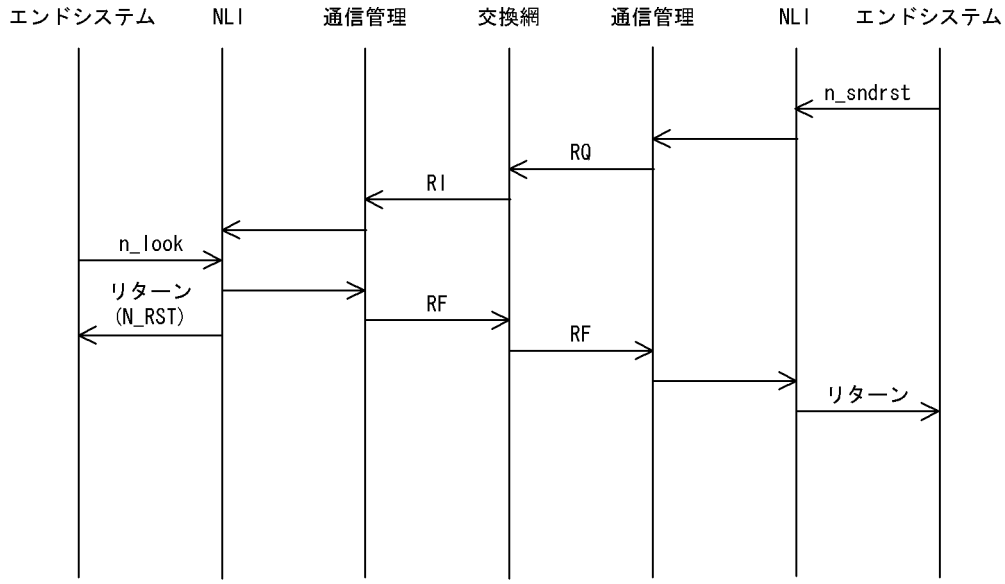
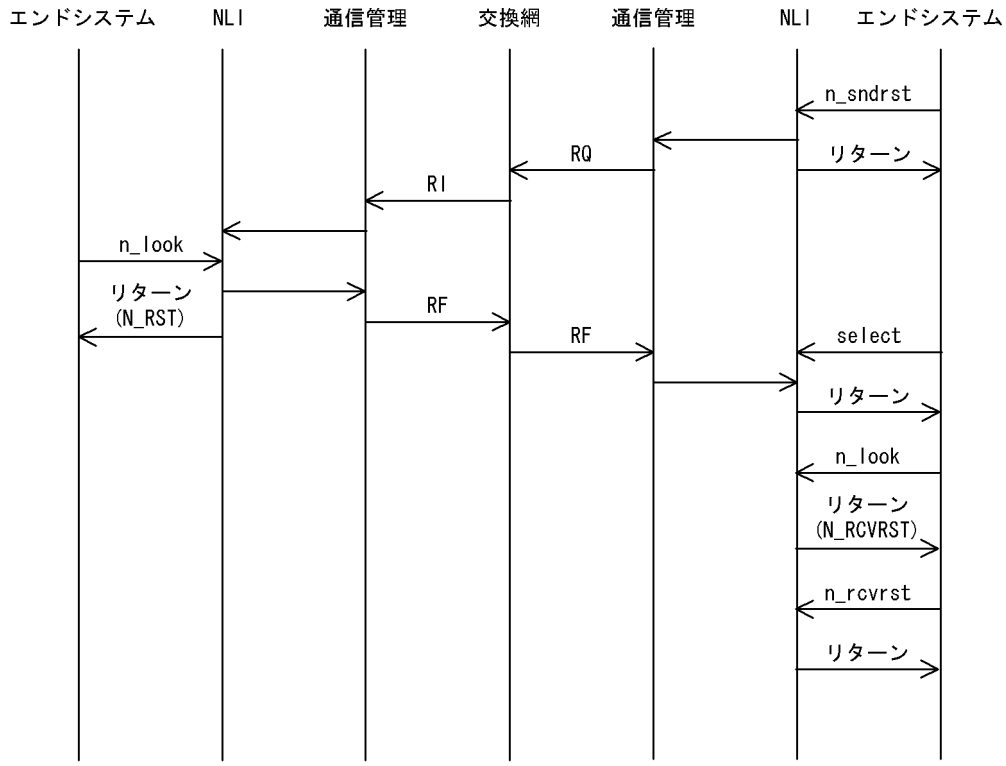


図 3-13　非同期型のネットワークコネクションのリセット手順



3.7 優先データ受信

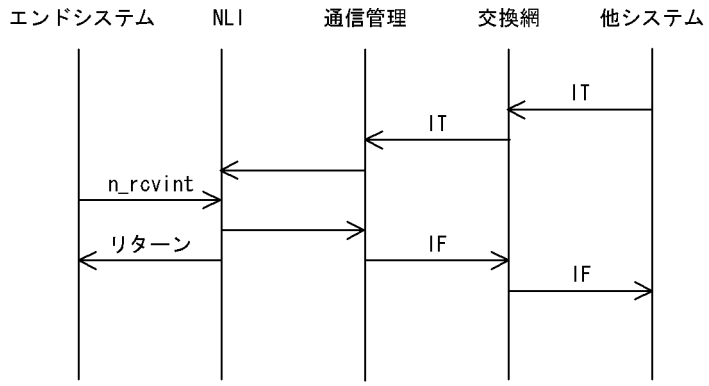
優先データの受信フェーズでは，確立したネットワークコネクションを使用して，n_rcvint 関数で優先データの受信を行います。なお，優先データの送信機能はサポートしていません。

優先データ受信で使用する関数を表 3-6 に，優先データ受信の手順を図 3-14 に示します。

表 3-6 優先データ受信で使用する関数

関数	説明
n_rcvint	ネットワークコネクションを介して優先データを受信します。

図 3-14 優先データ受信の手順



3.8 状態遷移

ネットワークインタフェースは，次の二つの構成要素を持ちます。

- ネットワークサービスをユーザに提供するライブラリの各関数
- 各関数を呼び出せる，シーケンスを定義する状態遷移の規則

状態遷移の規則は，状態遷移図を使って説明します。状態遷移図では，状態情報とイベントの処理に基づいてライブラリコールの有効なシーケンスを定義します。

3.8.1 ネットワークインタフェース状態

ネットワークインタフェースの状態遷移の説明で使う状態を定義します。ネットワークインタフェース状態を表 3-7 に示します。

表 3-7 ネットワークインタフェース状態

状態	説明
N_UNINIT	初期状態（インタフェースの初期化および最終状態）
N_UNBIND	初期化されているが，結合されていない状態
N_IDLE	接続が確立されていない状態
N_OUTCON	クライアント用の発信接続の保留中の状態
N_INCON	サーバ用の着信接続の保留中の状態
N_DATAXFER	データ転送ができる状態

3.8.2 発信イベント

ネットワークインタフェースの状態遷移の説明で使う発信イベントを定義します。ネットワークインタフェース発信イベントを表 3-8 に示します。

表 3-8 ネットワークインタフェース発信イベント

発信イベント	説明
opened	n_open の正常リターン
bind	n_bind の正常リターン
bind2	n_bind2 の正常リターン
unbind	n_unbind の正常リターン
closed	n_close の正常リターン
connect1	同期モードで，n_connect の正常リターン
connect2	非同期モードで，n_connect の NNODATA リターン
accept1	ocnt==1，fd==resfd，n_accept の正常リターン

発信イベント	説明
accept2	ocnt==1，fd!=resfd，n_accept の正常リターン
accept3	ocnt>1，n_accept の正常リターン
snd	n_snd の正常リターン
snddis1	ocnt<=1，n_snddis の正常リターン
snddis2	ocnt>1，n_snddis の正常リターン
sndrst	n_sndrst の正常リターン
rcvrst	n_rcvrst の正常リターン
rcvint	n_rcvint の正常リターン
rstrsp	n_rstrsp の正常リターン

（凡例）

ocnt：未処理接続指示の数

fd：現在のネットワーク端点のファイル記述子

resfd：接続が受け付けられるネットワーク端点のファイル記述子

3.8.3 着信イベント

ネットワークインタフェースの状態遷移の説明で使う着信イベントを定義します。ネットワークインタフェース着信イベントを表 3-9 に示します。

表 3-9 ネットワークインタフェース着信イベント

着信イベント	説明
listen	n_listen の正常リターン
rcvconnect	n_rcvconnect の正常リターン
rcv	n_rcv の正常リターン
rcvdis1	ocnt<=0，n_rcvdis の正常リターン
rcvdis2	ocnt==0，n_rcvdis の正常リターン
rcvdis3	ocnt>0，n_rcvdis の正常リターン
pass_conn	渡された接続の受け付け

（凡例）

ocnt：未処理接続指示の数

3.8.4 ネットワークインタフェースの状態遷移

ネットワークインタフェースの状態遷移（VC 用）を表 3-10 に，ネットワークインタフェースの状態遷移（PVC 用）を表 3-11 に示します。また，ネットワークインタフェー

スの状態遷移の代表的な例を図 3-15 に示します。

表 3-10　ネットワークインタフェースの状態遷移（VC 用）

イベント	状態					
	N_UNINIT (0)	N_UNBIND (1)	N_IDLE (2)	N_OUTCON (3)	N_INCON (4)	N_DATAXFER (5)
opened	→ (1)					
bind		→ (2) ※ 1				
unbind			→ (1)			
closed		→ (0)	→ (0)	→ (0)	→ (0)	→ (0)
connect1			→ (5)			
connect2			→ (3)			
rcvconnect				→ (5)		
listen			→ (4)			
accept1					→ (5) ※ 3	
accept2					→ (2) ※ 3, ※ 4	
accept3					＿※ 3, ※ 4	
snd						—
rcv						—
snddis1				→ (2)	→ (2)	→ (2)
snddis2					＿※ 3	
rcvdis1				→ (2)	→ (2)	→ (2)
rcvdis2					→ (2) ※ 3	
rcvdis3					＿※ 3	
pass_conn			→ (5)			
rcvint						—
bind2		→ (2) ※ 1				
rstrsp						—

（凡例）

－：状態は遷移しません。

空白：該当しません。

注

n_alloc，n_free，n_error，n_getstate，n_getinfo，n_look，および n_sync の各関

数は，N_UNBIND 以降のどんな状態でも発行できるため，含めていません。また，n_chglist 関数は，N_IDLE の場合はどんな状態でも発行できるため，含めていません。

注※ 1
未処理の接続指示の数をゼロに設定します。

注※ 2
未処理の接続指示の数を一つ増やします。

注※ 3
未処理の接続指示の数を一つ減らします。

注※ 4
接続を n_accept で示す別のネットワーク端点に移します。

表 3-11　ネットワークインタフェースの状態遷移（PVC 用）

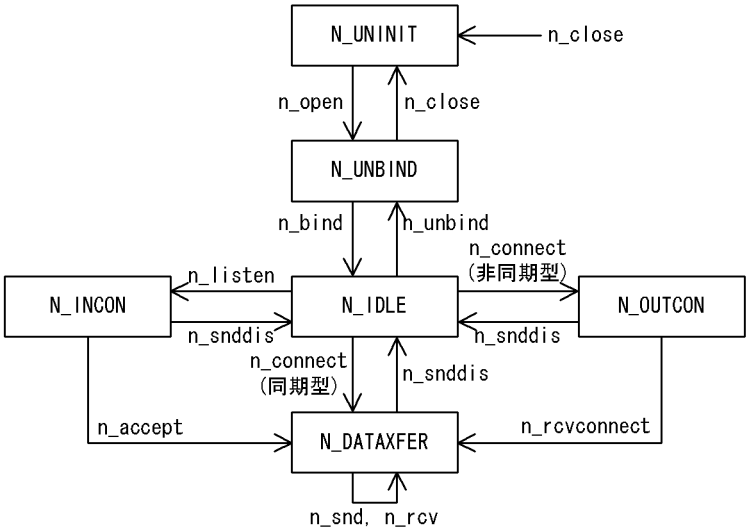
イベント	状態					
	N_UNINIT (0)	N_UNBIND (1)	N_IDLE (2)	N_OUTCON (3)	N_INCON (4)	N_DATAXFER (5)
opened	→ (1)					
bind		→ (2)				
unbind			→ (1)			
closed		→ (0)	→ (0)	→ (0)	→ (0)	→ (0)
connect1			→ (5)			
connect2			→ (3)			
rcvconnect				→ (5)		
listen			→ (4)			
accept1					→ (5)	
snd						—
rcv						—
snddis				→ (2)		→ (2)
rcvdis				→ (2)	→ (2)	→ (2)
sndrst						—
rcvrst						—
rstrsp						—

（凡例）
—：状態は遷移しません。
空白：該当しません。

注

n_alloc，n_free，n_error，n_getstate，n_getinfo，n_look，および n_sync の各関数は，N_UNBIND 以降のどんな状態でも発行できるため，含めていません。

図 3-15　ネットワークインタフェースの状態遷移の代表的な例



4 NLI のインタフェース

この章では，NLI のインタフェースについて説明します。

4.1 インタフェース

NLI の API（アプリケーションプログラムインタフェース）は，次に示すように二つの部分から構成されます。

- インタフェース構造体
- ライブラリ関数

インタフェース構造体は，データ送受信を除く手順で NLI と上位プログラムとのインタフェース情報を受け渡しするための領域の定義であり，あらかじめ型枠を決めている構造体です。上位プログラムは，ライブラリ関数を呼び出す前に，その関数で必要な情報をインタフェース構造体に設定しておきます。ライブラリ関数については，「5. ライブラリ関数」を参照してください。

NLI からのリターン情報は，関数から戻ってきたときにインタフェース構造体に書き込まれています。

4.2 インタフェースの構造体

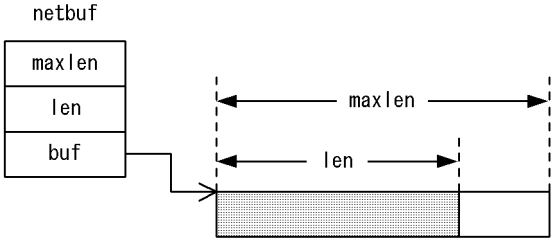
NLI では，NLI 関数とユーザプログラム間でのやり取りを行うために，データ構造体を定義しています。NLI で使用する構造体一覧を表 4-1 に示します。

表 4-1 NLI で使用する構造体一覧

構造体名称	属性	要素名称
struct netbuf ※ 1	unsigned int	maxlen
	unsigned int	len
	char *	buf
struct n_info ※ 2	long	addr
	long	options
	long	nsdu
	long	ensdu
	long	connect
	long	discon
	long	servtype
struct n_bind	struct netbuf	addr
	unsigned int	qlen
struct n_discon	struct netbuf	udata
	int	reason
	int	sequence
struct n_call	struct netbuf	addr
	struct netbuf	opt
	struct netbuf	udata
	int	sequence
struct n_reset	unsigned char	ver
	unsigned char	cause
	unsigned char	diag
	unsigned char	cdgflg

注※ 1

netbuf の構造体を次に示します。



buf 要素がユーザのバッファを指し示し，バッファ中のデータ量を len 要素で示します。ユーザの buf 要素に情報を書き込む場合は，バッファ領域を超える書き込みを防ぐために，maxlen 要素がバッファの最大長を示します。netbuf 構造体で参照だけをさせる場合は，maxlen は無視されます。

注※ 2

「5.3.14　n_open」のネットワーク端点の情報を参照してください。

5 ライブラリ関数

この章では，ライブラリ関数の文法について説明します。

5.1 ライブラリ関数の一覧

NLI が提供するライブラリ関数の一覧を表 5-1 に示します。なお，インクルードファイルは，<xnfw/niuser.h> となります。

表 5-1 ライブラリ関数の一覧

関数名	機能概要
n_accept	ネットワーク接続に対する要求を受け入れます。
n_alloc	ネットワークインタフェースデータ構造体を割り当てます。
n_bind	ネットワーク端点にネットワークアドレスを結合します。
n_bind2	相手局ネットワークアドレスを指定して，ネットワークアドレスを結合します。
n_chglist	相手局ネットワークアドレスを変更します。
n_close	ネットワーク端点をクローズします。
n_connect	指定されたあて先でネットワークユーザとの接続を確立します。
n_error	ネットワークインタフェースエラーメッセージを表示します。
n_free	n_alloc で割り当てられた構造体を解放します。
n_getinfo	特定のネットワークプロバイダに関する一連の情報を返します。
n_getstate	ネットワーク端点の状態を返します。
n_listen	ネットワークユーザからの接続要求の指示を待ちます。
n_look	ネットワーク端点の現イベントを返します。
n_open	ネットワークプロバイダに接続されているネットワーク端点を確立します。
n_rcv	ネットワークコネクションを介してデータを受信します。
n_rcvconnect	非同期モードで n_connect を呼び出したあとに接続の確立を通知します。
n_rcvdis	ネットワークコネクションの解放指示を受信します。
n_rcvint	ネットワークコネクションを介して優先データを受信します。
n_rcvrst	ネットワークコネクションのリセット情報を取得します。
n_rstrsp	ネットワークコネクションのリセットを応答します。
n_snd	ネットワークコネクションを介してデータを送信します。
n_snddis	ネットワークコネクションの解放を要求します。
n_sndrst	ネットワークコネクションをリセットします。
n_sync	ネットワークプロバイダとネットワーク端点を同期化します。
n_unbind	ネットワーク端点からネットワークアドレスの結合を解除します。

5.2 記述形式

ライブラリ関数は，次のような形式で説明しています。ただし，必ずしもすべての項目を含んでいるとは限りません。

(1) 機能概要

NLI が提供するライブラリ関数の機能概要を記述しています。

(2) 構文

ライブラリ関数の構文を C 言語を使用して記述しています。

(3) 設定情報

ライブラリ関数の引数について説明しています。

(4) リターン情報

NLI がアプリケーションに報告する内容を説明しています。

(a) リターン情報

リターンコードの種類とその意味を説明しています。

(b) 詳細エラーコード

異常終了発生時の詳細なエラーコードとその意味を説明しています。

(5) 特記事項

ライブラリ関数に関する特記事項を説明しています。

5.3 ライブラリ関数の詳細

5.3.1 n_accept

(1) 機能概要

ネットワーク端点に到着したコネクション確立指示を受諾します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_accept(fd, resfd, call);
int rcode, fd;
int resfd;
struct n_call *call;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) resfd

コネクション確立指示を受諾するネットワーク端点を指定します。ただし，resfd!=fd の場合，resfd のネットワーク端点は N_IDLE 状態でなければなりません。

(c) call

コネクションを確立するために必要な情報を指定します。

n_call 構造体の設定内容については,「表 10-10　インタフェース構造体（n_call）のパラメタの設定・参照（その 1)」を参照してください。

● VC の場合

addr：

n_listen で受け取ったコネクション確立要求側のネットワークアドレスを指定します。

opt：

ネットワークプロトコルオプションを指定します。ただし，パケットサイズは，コネクション確立要求側の値および自局の構成定義で設定した値以下でなければなりません。優先データの使用については,「11.1　優先データ受信機能」を参照してください。

udata：

コールユーザデータを指定します。コールデータユーザの使用については,「11.2

コールユーザデータの転送機能」を参照してください。

sequence：

n_listen で受け取った値を指定します。

● PVC の場合

設定しても，NLI で無視されます。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

(5) 特記事項

fd=resfd とした場合，そのネットワーク端点では，二度とコネクション確立指示を受け付けることができません。

5.3.2 n_alloc

(1) 機能概要

ネットワーク関数の引数となる構造体に対してメモリを割り当てます。また，その構造体が参照するバッファに対してもメモリを割り当てます。

(2) 構文

```
#include <xnfw/niuser.h>
#include <stdio.h>
raddr = *n_alloc(fd, struct_type, fields);
char *raddr;
int fd;
int struct_type;
int fields;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) struct_type

割り当てる構造体のタイプを指定します。割り当てる構造体のタイプを表 5-2 に示します。

表 5-2 割り当てる構造体のタイプ

タイプ	割り当てる構造体
N_BIND	struct n_bind
N_CALL	struct n_call
N_DIS	struct n_discon
N_INFO	struct n_info
N_RESET	struct n_reset

(c) fields

struct_type で指定した構造体に関する，すべてのフィールドを割り当てる N_ALL を指定します。

fields に上記以外の値が指定された場合，n_errno に NEPARAM を設定してエラーリターンします。

(4) リターン情報

(a) リターンコード

整数値：新たに割り当てた構造体のポインタ

0：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

(5) 特記事項

buf ポインタおよび maxlen は，n_open および n_getinfo 関数でユーザに返される大きさの情報に基づいて初期設定されます。ただし，指定したフィールドに関連する値が -1 または -2 の場合，指定されていないフィールドには，buf に NULL が，maxlen に 0 が設定されます。

5.3.3 n_bind

(1) 機能概要

ネットワーク端点にネットワークアドレスを結合します。このあと，ネットワークプロバイダはネットワーク端点でコネクション確立指示の受諾やコネクション確立要求を行うことができます。PVC 接続では，ネットワーク端点と LCI（論理チャネル情報）を対応づけます。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_bind(fd, req, ret);
int rcode, fd;
struct n_bind *req;
struct n_bind *ret;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) req

ネットワークプロトコル提供者と受諾するために必要な情報を指定します。

n_bind 構造体の設定内容については，「表 10-5　インタフェース構造体（n_bind）のパラメタの設定・参照（その 1)」を参照してください。

● VC の場合

addr：

設定しても，NLI で無視されます。

qlen：

処理できるコネクション確立指示の数を指定します。
qlen=0 指定時，クライアントとして動作します。以後，n_connect を発行してください。
qlen!=0 指定時，サーバとして動作します。以後，n_listen を発行してください。
qlen!=0 のネットワーク端点は，システムで一つだけ存在できます。

● PVC の場合

addr：

LCI などの情報を指定します。ここに設定する情報形式については「図 7-2　アドレスの形式（PVC 接続）」を参照してください。

qlen：

設定しても，NLI で無視されます。

(c) ret

ネットワークプロトコル提供者の受諾した情報を受け取るエリアを指定します。

n_bind 構造体の参照内容については,「表 10-5　インタフェース構造体（n_bind）のパラメタの設定・参照（その 1)」を参照してください。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は,「9.1　エラー番号一覧」を参照してください。

n_errno に NCOMERR が設定された場合，n_comerr に詳細エラーコードが設定されます。詳細エラーコードについては，マニュアル「通信管理　XNF/AS　解説・運用編」を参照してください。

(b) ret

● VC の場合

addr：
　設定されません。

qlen：
　受諾された，処理できるコネクション確立指示の数が設定されます。

● PVC の場合

addr：
　受諾された，LCI などの情報が設定されます。ここに設定される情報形式については「図 7-2　アドレスの形式（PVC 接続)」を参照してください。

qlen：
　設定されません。

(5) 特記事項

この関数を発行した qlen!=0 のネットワーク端点は，どんな相手局からのコネクション確立指示でも受け付けられます。n_bind2 関数を発行した qlen!=0 のネットワーク端点は，n_bind2 関数で指定した相手局ネットワークアドレスと一致した相手局からのコネクション確立指示だけ受け付けられます。

5.3.4 n_bind2

(1) 機能概要

VC 接続で，ネットワーク端点にネットワークアドレスを結合します。このあと，指定した相手局ネットワークアドレスと一致したコネクション確立指示を，このネットワーク端点で受諾します（上位プロセス選択機能）。

n_bind を発行した qlen!=0 のサーバ端点は，受信したコネクション確立指示がどんな相手局からでも受諾しますが，この関数を発行した qlen!=0 のサーバ端点は，指定した相手局ネットワークアドレスと一致したコネクション確立指示だけ受諾できます。

コネクション確立指示受信時，n_bind と n_bind2 の qlen!=0 の両方のサーバ端点がある場合，n_bind2 のネットワークアドレスとのチェックをして，一致すれば n_bind2 のサーバ端点に通知します。一致しなければ，n_bind のサーバ端点に通知します。

コネクション確立指示受信時，n_bind2 の qlen!=0 のサーバ端点が複数あり，複数のサーバが，指定した相手局ネットワークアドレスと一致する場合，どのサーバ端点に通知するかは保証できません。

qlen!=0 のサーバ端点は，n_bind の場合，1 個存在できます。n_bind2 の場合，次のようになります。

- XNF/AS/NLI V1 製品の場合
  8 個まで存在できます。
- XNF/AS/NLI V2 製品の場合
  configuration 文の max_NLI_VC_server オペランドの値まで存在できます。
  max_NLI_VC_server オペランドの説明の詳細は，マニュアル「通信管理　XNF/AS 構成定義編」を参照してください。

上位プロセス選択機能を使用する場合，n_open 発行時に，oflag に O_XNFNLI_MLT を設定する必要があります。

(2) 構文

```
#include <xnfw/niuser.h>
rcode= n_bind2(fd, req, ret);
int  rcode, fd;
struct n_bind  *req;
struct n_bind  *ret;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) req

ネットワークプロトコル提供者と受諾するために必要な情報を指定します。

n_bind 構造体の設定内容については,「表 10-6　インタフェース構造体（n_bind）のパラメタの設定・参照（その 2)」を参照してください。

addr：

相手局のネットワークアドレスを指定します。ネットワークアドレスについては,「7.1　ネットワークアドレス」を参照してください。

qlen：

処理できるコネクション確立指示の数を指定します。
qlen=0 指定時，クライアントとして動作します。以後，n_connect を発行してください。
qlen!=0 指定時，サーバとして動作します。以後，n_listen を発行してください。
qlen!=0 のネットワーク端点は，次のようになります。

- XNF/AS/NLI V1 製品の場合
  システムで 8 個まで存在できます。
- XNF/AS/NLI V2 製品の場合
  システムで configuration 文の max_NLI_VC_server オペランドの値まで存在できます。
  max_NLI_VC_server オペランドの説明の詳細は，マニュアル「通信管理　XNF/AS　構成定義編」を参照してください。

(c) ret

ネットワークプロトコル提供者の受諾した情報を受け取るエリアを指定します。

ret が NULL の場合，何も設定されません。

n_bind 構造体の参照内容については,「表 10-6　インタフェース構造体（n_bind）のパラメタの設定・参照（その 2)」を参照してください。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は,「9.1　エラー番号一覧」を参照してください。

(b) ret

addr：

設定されません。

qlen：
受諾された，処理できるコネクション確立指示の数が設定されます。

5.3.5 n_chglist

(1) 機能概要

VC 接続で，n_bind2 で指定した相手局ネットワークアドレスを変更します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode= n_chglist(fd, chgbind);
int  rcode, fd;
struct n_bind  *chgbind;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) chgbind

変更したい相手局ネットワークアドレスを指定します。

n_bind 構造体の設定内容については，「表 10-7　インタフェース構造体（n_bind）のパラメタの設定・参照（その 3)」を参照してください。

addr：

相手局ネットワークアドレスを指定します。ネットワークアドレスについては，「7.1　ネットワークアドレス」を参照してください。

qlen：

設定しても，NLI で無視されます。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

5.3.6 n_close

(1) 機能概要

ネットワーク端点をクローズします。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_close(fd);
int rcode, fd;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は,「9.1 エラー番号一覧」を参照してください。

(5) 特記事項

n_alloc で確保した領域は解放しません。

5.3.7 n_connect

(1) 機能概要

ほかのネットワークユーザに対して，コネクション接続を要求します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_connect(fd, sndcall, rcvcall);
int rcode, fd;
struct n_call *sndcall;
struct n_call *rcvcall;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) sndcall

コネクションを確立するために必要な情報を指定します。

n_call 構造体の設定内容については，「表 10-11　インタフェース構造体（n_call）のパラメタの設定・参照（その 2)」を参照してください。

● VC の場合

addr：

相手のネットワークアドレス情報を指定します。ネットワークアドレスについては，「7.1　ネットワークアドレス」を参照してください。

opt：

ネットワークプロトコルオプションを指定します。ただし，パケットサイズは，自局の構成定義で設定した値以下でなければなりません。優先データの使用については，「11.1　優先データ受信機能」を参照してください。

udata：

コールユーザデータを指定します。コールユーザデータの使用については，「11.2　コールユーザデータの転送機能」を参照してください。

sequence：

設定しても，NLI で無視されます。

● PVC の場合

設定しても，NLI で無視されます。

(c) rcvcall

コネクションの情報を受け取るエリアを指定します。

n_call 構造体の参照内容については,「表 10-11　インタフェース構造体（n_call）のパラメタの設定・参照（その 2)」を参照してください。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は,「9.1　エラー番号一覧」を参照してください。

(b) rcvcall

● VC の場合

addr：
　コネクション確立着信側のネットワークアドレスが設定されます。

opt：
　コネクション確立着信側が指定したネットワークプロトコルオプションが設定されます。プロトコルオプションとして，パケットサイズと優先データ使用有無が設定されます。プロトコルオプションについては,「8.　ネットワークプロトコルオプション」を参照してください。

udata：
　コールユーザデータが設定されます。コールユーザデータの使用については,「11.2　コールユーザデータの転送機能」を参照してください。

sequence：
　設定されません。

● PVC の場合

addr：
　LCI などの情報が設定されます。ここに設定される情報の形式については,「7.1　ネットワークアドレス」を参照してください。

opt：
　設定されません。

udata：
　設定されません。

sequence：
　設定されません。

(5) 特記事項

1. 非同期モードで動作している場合，n_connect は遠隔ユーザの応答を待たないですぐ

にリターンします。その場合，-1 でリターンして，n_errno には NNODATA が設定されます。同期モードで動作している場合，相手のネットワークユーザの応答を待ってからリターンします。

2. udata フィールドの１バイト目は，通信システム内で調整されているプロトコル識別子（SPI）を指定します。

5.3.8 n_error

(1) 機能概要

関数実行中に生じた，エラー番号に対応するエラーメッセージを標準エラー出力に出力します。

(2) 構文

```
#include <xnfw/niuser.h>
extern int n_errno;
extern char *n_errlist[];
extern int n_err;
void n_error(errmsg);
char *errmsg;
```

(3) 設定情報

(a) errmsg

ユーザが出力したい文字列を指定します。

(4) リターン情報

標準エラー出力にエラーメッセージを出力します。

(5) 特記事項

1. n_errlist は，エラーに対応するメッセージ文字列の配列であり，n_errno をインデックスとして使用します。
2. n_err は，n_errlist テーブルで提供されるメッセージの最大番号です。
3. n_errno は，エラーが発生したときだけエラー番号が設定され，クリアされません。

(6) 表示例

n_connect 関数が，ネットワーク端点 fd2 で不正なアドレスが与えられたため，異常終了（n_errno=NBADADDR）し，続いて次のようにこの関数が呼ばれたとします。

```
n_errno(" n_connect failed on fd2 ");
```

このとき，出力される診断メッセージは次のようになります。

```
n_connect failed on fd2：incorrect address format
```

エラーメッセージ一覧を表 5-3 に示します。

表 5-3 エラーメッセージ一覧

エラー番号	エラーメッセージ
NBADF	illegal network fd
NOUTSTATE	out of state
NNODATA	no data
NNOTSUPPORT	primitive/action not supported
NBADOPT	incorrect option format
NBADFLAG	incorrect flag
NBADSEQ	bad call sequence number
NBADDATA	illegal amount of data
NBADADDR	incorrect addr format
NACCESS	incorrect permissions
NBUFOVFLW	buffer not large enough
NFLOW	flow control
NLOOK	event requires attention
NNODIS	discon_ind not found on queue
NSTATECHNG	state is in process of changin
NSYSERR	system error
NCOMERR	communication error
NEPARAM	parameter error
NOVRCON	over permitted connection number

5.3.9 n_free

(1) 機能概要

n_alloc によって割り当てられた構造体のメモリを解放します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_free(ptr, struct_type);
int rcode, struct_type;
char *ptr;
```

(3) 設定情報

(a) ptr

解放する構造体のアドレスを指定します。

(b) struct_type

「5.3.2 n_alloc」を参照してください。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は,「9.1 エラー番号一覧」を参照してください。

5.3.10　n_getinfo

（1） 機能概要

現在のネットワーク端点の情報を取得します。

（2） 構文

```
#include <xnfw/niuser.h>
rcode = n_getinfo(fd, info);
int rcode, fd;
struct n_info *info;
```

（3） 設定情報

(a) fd

ネットワーク端点を指定します。

(b) info

情報を受け取るエリアを指定します。

（4） リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

(b) info

info 構造体には，n_open で返された情報と同じ情報が設定されます。詳細は，表 5-7 および表 5-8 を参照してください。

5.3.11 n_getstate

(1) 機能概要

ネットワーク端点に関連する，ネットワークプロトコル提供者の現在の状態を取得します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_getstate(fd);
int rcode, fd;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(4) リターン情報

(a) リターンコード

正数：正常終了（現在の状態を示しています。表 5-4 を参照してください）

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

表 5-4　ネットワーク端点に関するプロバイダ状態一覧

状態	意味
N_UNBIND	結合されていません。
N_IDLE	アイドル状態です。
N_OUTCON	出ていく接続が保留中です。
N_INCON	入ってくる接続が保留中です。
N_DATAXFER	データ転送ができます。

5.3.12 n_listen

(1) 機能概要

ほかのネットワークユーザからのコネクション確立指示の到着を待ちます。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_listen(fd, call);
int rcode, fd;
struct n_call *call;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) call

コネクションの情報を受け取るエリアを指定します。

n_call 構造体の参照内容については,「表 10-10　インタフェース構造体（n_call）のパラメタの設定・参照（その 1)」を参照してください。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は,「9.1　エラー番号一覧」を参照してください。

(b) call

● **VC の場合**

addr：

コネクション確立着信側のネットワークアドレスが設定されます。

opt：

コネクション確立着信側が指定したネットワークプロトコルオプションが設定されます。プロトコルオプションとして，パケットサイズと優先データ使用有無が設定されます。プロトコルオプションについては,「8.　ネットワークプロトコルオプション」を参照してください。

udata：

コールユーザデータが設定されます。コールユーザデータの使用については，「11.2　コールユーザデータの転送機能」を参照してください。

sequence：

コネクション確立指示を一意に認識するための番号が設定されます。この番号を，n_accept または n_snddis で指定します。

● PVC の場合

addr：

LCI などの情報が設定されます。ここに設定される情報の形式については，「7.1　ネットワークアドレス」を参照してください。

opt：

設定されません。

udata：

設定されません。

sequence：

設定されません。

(5) 特記事項

n_listen は，非同期モードで動作している場合，相手からのコネクション接続指示が到着していないと -1 でリターンし，n_errno に NNODATA が設定されます。同期モードで動作している場合，相手からのコネクション接続指示が到着するまでリターンしません。

5.3.13　n_look

(1) 機能概要

ネットワーク端点上での現在の発生イベントを返します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_look(fd);
int rcode, fd;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(4) リターン情報

(a) リターンコード

正数：正常終了（発生したイベントを示しています。表 5-5 を参照してください）

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

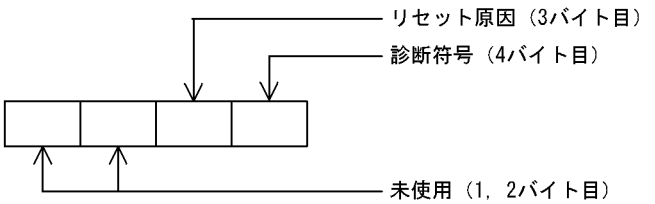
表 5-5　ネットワーク端点で発生したイベント一覧

イベント	意味	対処
0	イベントなし	特になし
N_LISTEN	コネクション確立指示の受信	n_listen を発行する
N_CONNECT	コネクション確立確認の受信	n_rcvconnect を発行する
N_DATA	ユーザデータの受信	n_rcv を発行する
N_DISCONNECT	コネクション解放通知の受信	n_rcvdis を発行する
N_RST	コネクションリセット要求の受信	n_comerr を参照する
N_RST2	コネクションリセット要求の受信	n_comerr を参照し， n_rstrsp を発行する
N_RCVRST	コネクションリセット確認の受信	n_rcvrst を発行する
N_INT	優先データの受信	n_rcvint を発行する

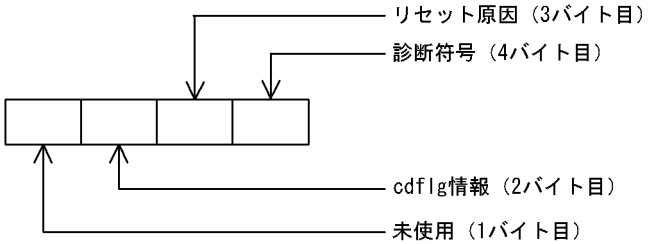
(5) 特記事項

1. この関数で N_RST が返された場合，グローバル変数 n_comerr に，リセット原因，

および診断符号が，次の形式で設定されます。



2. この関数で N_RST2 が返された場合，グローバル変数 n_comerr に，cdflg 情報，リセット原因，および診断符号が，次の形式で設定されます。



cdflg 情報に設定されるコードと意味を次に示します。

- 0x00：相手局からリセット指示を受信
- 0x03：通信管理の内部リセット発生

3. ユーザへの即時の通知を必要とするイベントには，現在実行中の関数，または次に実行される関数で，n_errno に NLOOK が設定されます。この場合，この関数を発行してください。
4. データ分割送信中，またはデータ分割受信中に N_RST が通知された場合，リセットが交換機でデータを追い越す可能性があります。そのため，意図したとおりにデータの組み立てができない場合があるので，該当するデータは破棄してください。
   データ分割送信中：n_snd のリターンが，要求長より小さい，または N_MORE を設定して送信している。
   データ分割受信中：n_rcv のリターンで，N_MORE が設定されている。

5.3.14 n_open

(1) 機能概要

未使用のネットワーク端点を開設し，そのファイル記述子を返します。

(2) 構文

```
#include <xnfw/niuser.h>
#include <fcntl.h>
#include <stdio.h>
rcode = n_open(path, oflag, info);
char *path;
int rcode, oflag;
struct n_info *info;
```

(3) 設定情報

(a) path

ネットワークプロトコル提供者名を指定します。

VC 接続の場合，“ X25-VC “ を指定します。PVC 接続の場合，” X25-PVC “ を指定します。

これら以外の値を指定した場合は，n_errno に NEPARAM が設定されて，エラーリターンします。

(b) oflag

ネットワーク端点の属性を指定します。

表 5-6 の属性を論理和形式で設定してください。ただし，同期モードと非同期モードは同時に指定できません。

表 5-6 oflag に設定するパラメタ一覧

パラメタ	意味	使用サービス
NULL	同期モードの実行	VC，PVC
O_NDELAY	非同期モードの実行（n_snddis を除きます）	VC，PVC
O_XNFNLI_DISC	n_snddis の非同期モードの実行	VC
O_XNFNLI_INT	優先データサービスの使用	VC
O_XNFNLI_CALL	コールユーザデータサービスの使用	VC
O_XNFNLI_QBIT	Q ビットインタフェースの使用（PVC サービスの場合はデフォルトで使用できます）	VC
O_XNFNLI_LCN	論理チャネル情報通知機能の使用	VC
O_XNFNLI_MLT	上位プロセス選択機能の使用	VC

パラメタ	意味	使用サービス
O_XNFNLI_NSAP	NSAP アドレス付加機能	VC
O_XNFNLI_RSTRSP	RF パケット応答同期機能	VC，PVC

(c) info

ネットワーク端点に関する，情報を設定するエリアを指定します。

(4) リターン情報

(a) リターンコード

0 以上：正常終了（ファイル記述子（ネットワーク端点）を返します）

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

(b) info

info で指定したエリアに開設した，ネットワーク端点に関する情報が設定されます。

info のそれぞれのフィールドに設定される値を表 5-7 および表 5-8 に示します。

表 5-7　X.25 プロトコルの VC サービスを使用する場合

フィールド名	値	意味
addr	50 または 56	ネットワークプロトコルアドレスの最大サイズ※1
options	4 または 8	ネットワークプロトコルオプションの最大サイズ※2
nsdu	-1	送信のサイズは無制限
ensdu	-2	優先データの送信を未サポート
connect	16	接続確立関数の送信データの最大サイズ
discon	16	途中解放関数の送信データの最大サイズ
servtype	N_CONS	コネクションモードのサービスをサポート

注※ 1

oflag で O_XNFNLI_LCN を指定すると，56 を返します。

注※ 2

oflag で O_XNFNLI_INT を指定すると，8 を返します。

表 5-8 X.25 プロトコルの PVC サービスを使用する場合

フィールド名	値	意味
addr	26	ネットワークプロトコルアドレスの最大サイズ
options	-2	ネットワークプロトコルオプションを未サポート
nsdu	-1	送信のサイズは無制限
ensdu	-2	優先データの送信を未サポート
connect	-2	接続確立関数の送信データを未サポート
discon	-2	途中解放関数の送信データを未サポート
servtype	N_PVC	PVC サービスをサポート

5.3.15 n_rcv

(1) 機能概要

データを受信します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_rcv(fd, buf, nbytes, flags);
int rcode, fd;
char *buf;
unsigned int nbytes;
int *flags;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) buf

データを受信するバッファアドレスを指定します。

(c) nbytes

受信バッファのサイズを指定します。

(d) flags

受信データの付加情報が設定されるエリアを指定します。

(4) リターン情報

(a) リターンコード

0 以上：正常終了（受信したデータ長を返します）

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

(b) 受信データの付加情報（flags）

flags で指定したエリアに，次の受信データの付加情報が設定されます。なお，受信パケットの内容によっては，N_MORE と N_QBIT には論理和形式で設定される場合があります。

N_MORE：
　継続するデータがあり，さらに n_rcv の発行が必要であることを示します。

N_QBIT：

受信データに，Q ビットが付加されていることを示します。

NULL：

継続するデータはなく，かつ Q ビットも付加されていないことを示します。

(5) 特記事項

n_rcv は，非同期モードで動作している場合はデータ未受信であれば，-1 でリターンして，n_errno に NNODATA が設定されます。また，データを受信しているが後続データがある場合，すでに受信したデータだけをリターンします。同期モードで動作している場合，データ未受信であれば，データを受信するまでリターンしません。また，データを受信しているが後続データがある場合は，受信バッファが満杯になるまでリターンしません。

5.3.16 n_rcvconnect

(1) 機能概要

非同期モードで発行した n_connect に対するコネクション確立確認を受信します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_rcvconnect(fd, call);
int rcode, fd;
struct n_call *call;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) call

コネクションの情報を受け取るエリアを指定します。

n_call 構造体の参照内容については,「表 10-10　インタフェース構造体（n_call）のパラメタの設定・参照（その 1)」を参照してください。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は,「9.1　エラー番号一覧」を参照してください。

(b) call

● VC の場合

addr：
　コネクション確立着信側のネットワークアドレスが設定されます。

opt：
　コネクション確立着信側が指定したネットワークプロトコルオプションが設定されます。プロトコルオプションとして，パケットサイズと優先データ使用有無が設定されます。プロトコルオプションについては,「8.　ネットワークプロトコルオプション」を参照してください。

udata：

コールユーザデータが設定されます。コールユーザデータの使用については，「11.2　コールユーザデータの転送機能」を参照してください。

sequence：

設定されません。

● PVC の場合

設定されません。

(5) 特記事項

相手からの確立確認を受信していない場合にこの関数を発行すると，-1 でリターンして，n_errno に NNODATA が設定されます。

5.3.17 n_rcvdis

(1) 機能概要

解放されたコネクションの原因コードを取得します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_rcvdis(fd, discon);
int rcode, fd;
struct n_discon *discon;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) discon

コネクション解放の情報を受け取るエリアを指定します。

n_discon 構造体の参照内容については,「表 10-8　インタフェース構造体（n_discon）のパラメタの設定・参照」を参照してください。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は,「9.1　エラー番号一覧」を参照してください。

(b) discon

● VC の場合

udata：

コネクション解放時の相手局からのユーザデータが設定されます。

reason：

コネクション解放時の切断原因および診断符号が設定されます。切断原因および診断符号については,「9.3　切断理由コード」を参照してください。

sequence：

コネクションが確立する前の n_listen で返された値が設定されます。

● PVC の場合

udata：

設定されません。

reason：

コネクション解放時の切断原因および診断符号が設定されます。切断原因および診断符号については,「9.3　切断理由コード」を参照してください。

sequence：

設定されません。

(5) 特記事項

NLI が生成する切断理由コードについては,「9.3　切断理由コード」を参照してください。

5.3.18 n_rcvint

(1) 機能概要

優先データを受信します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_rcvint(fd, buf, nbytes, flags);
int rcode, fd;
char *buf;
unsigned int nbytes;
int *flags;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) buf

優先データを受信するバッファアドレスを指定します。

(c) nbytes

受信バッファのサイズを指定します。

(d) flags

受信データの付加情報が設定されるエリアを指定します。

(4) リターン情報

(a) リターンコード

0 以上：正常終了（受信した優先データ長を返します）

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

(b) flags：受信データの付加情報

未使用。

(5) 特記事項

n_rcvint は，非同期モードで動作している場合，データ未受信であれば異常終了し，n_errno に NNODATA が設定されます。同期モードで動作している場合，データ未受信

であればデータを受信するまで関数はリターンしません。

5.3.19 n_rcvrst

(1) 機能概要

PVC で，リセット原因符号と診断符号を取得します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_rcvrst(fd,reset);
int rcode,fd;
struct n_reset *reset;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) reset

リセットされたコネクションの情報を受けるエリアを指定します。

n_reset 構造体の参照内容については，「表 10-9　インタフェース構造体（n_reset）のパラメタの設定・参照」を参照してください。

(4) リターン情報

(a) リターンコード

0 以上：正常終了（発生したイベントを示しています。表 5-9 を参照してください）

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

表 5-9　ネットワーク端点で発生したイベント一覧

イベント	意味
N_RSTIND	リセット指示受信
N_RSTCNF	リセット確認受信

(b) reset

ver：

　設定されません。

cause：

リセット原因符号が設定されます。

diag：
診断符号が設定されます。

cdgflg：
リセット情報が設定されます。

cdgflg に設定されるリセット情報を表 5-10 に示します。

表 5-10 cdgflg に設定されるリセット情報

値	意味
0	相手局からリセット確認を受信
1	相手局からリセット指示を受信
2	リセット送信がタイムアウト
3	リセット送信が通信管理の内部リセットと衝突

(5) 特記事項

データ分割送信中，またはデータ分割受信中にリセットが発生した場合，リセットが交換機でデータを追い越す可能性があります。そのため，意図したとおりにデータの組み立てができない場合があるので，該当するデータは破棄してください。

データ分割送信中：
n_snd のリターンが要求長より小さい，または N_MORE を設定して送信しています。

データ分割受信中：
n_rcv のリターンで N_MORE が設定されています。

5.3.20 n_rstrsp

(1) 機能概要

RF パケットを応答します。

(2) 構文

```
#include  <xnfw/niuser.h>
rcode = n_rstrsp(fd);
int rcode, fd;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は,「9.1 エラー番号一覧」を参照してください。

(5) 特記事項

「11.8 RF パケット応答同期機能」の注意事項を参照してください。

5.3.21 n_snd

(1) 機能概要

データを送信します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_snd(fd, buf, nbytes, flags);
int rcode, fd;
char *buf;
unsigned int nbytes;
int flags;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) buf

データを送信するバッファアドレスを指定します。

(c) nbytes

送信データのサイズを指定します。

(d) flags

送信データの付加情報を指定します。N_MORE と N_QBIT は，論理和形式で設定できます。

N_MORE：
　継続するデータがあり，ユーザがさらに n_snd を発行することを示します。

N_QBIT：
　送信するデータに，Q ビットを付加することを示します。

NULL：
　継続するデータはなく，かつ Q ビットも付加しないことを示します。

(4) リターン情報

(a) リターンコード

0 以上：正常終了（送信できたデータ長を返します）

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

(5) 特記事項

1. n_snd は，非同期モードで動作している場合に送信ビジーが発生すると，n_errno に NFLOW が設定され，実際に送信されたデータ長がリターンします。同期モードで動作している場合に送信ビジーが発生すると，指定したデータ送信が終了するまで，この関数はリターンしません。
2. 要求した送信データのサイズと，送信できたユーザデータ長（リターン値）は異なる場合があります。その場合，送信できなかった残りのデータを指定して，再度送信してください。

5.3.22 n_snddis

(1) 機能概要

コネクションの解放を要求します。また，コネクション確立指示の拒否応答を返します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_snddis(fd, call);
int rcode, fd;
struct n_call *call;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) call

コネクション解放に必要な情報を指定します。

n_call 構造体の設定内容については，「表 10-10　インタフェース構造体（n_call）のパラメタの設定・参照（その 1)」を参照してください。

● VC の場合

addr：
　設定しても，NLI で無視されます。

opt：
　設定しても，NLI で無視されます。

udata：
　コネクション確立の拒否または取り消しをするときに送る，ユーザデータを指定します。

sequence：
　コネクション確立を拒否するときに n_listen で返された値を指定します。

● PVC の場合

設定しても，NLI で無視されます。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は,「9.1　エラー番号一覧」を参照してください。

5.3.23 n_sndrst

(1) 機能概要

PVC で，接続を再初期化します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_sndrst(fd, req, ret);
int rcode, fd;
struct n_reset *req;
struct n_reset *ret;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(b) req

コネクションをリセットするために必要な情報を指定します。

n_reset 構造体の設定内容については，「表 10-9　インタフェース構造体（n_reset）のパラメタの設定・参照」を参照してください。

ver：
　設定しても，NLI で無視されます。

cause：
　リセット原因符号を指定してください。

diag：
　診断符号を指定してください。

cdgflg：
　設定しても，NLI で無視されます。

(c) ret

リセットされたコネクションの情報を受けるエリアを指定します。

n_reset 構造体の参照内容については，「表 10-9　インタフェース構造体（n_reset）のパラメタの設定・参照」を参照してください。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

（b） ret

ver：
　設定されません。

cause：
　リセット原因符号が設定されます。

diag：
　診断符号が設定されます。

cdgflg：
　リセット情報が設定されます。cdgflg の設定内容については，「表 5-10　cdgflg に設定されるリセット情報」を参照してください。

（5） 特記事項

データ分割送信中，またはデータ分割受信中にリセットが発生した場合，リセットが交換機でデータを追い越す可能性があります。そのため，意図したとおりにデータの組み立てができない場合があるので，該当するデータは破棄してください。

データ分割送信中：
　n_snd のリターンが要求長より小さい，または N_MORE を設定して送信しています。

データ分割受信中：
　n_rcv のリターンで N_MORE が設定されています。

5.3.24 n_sync

(1) 機能概要

ネットワーク端点の同期化を行います。また，ネットワーク端点に関連するネットワークプロトコル提供者の現在の状態を取得します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_sync(fd);
int rcode, fd;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(4) リターン情報

(a) リターンコード

正数：正常終了（現在の状態を示しています。「表 5-4　ネットワーク端点に関するプロバイダ状態一覧」を参照してください）

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1　エラー番号一覧」を参照してください。

(5) 特記事項

このプログラムでの n_sync は，n_getstate と同じ機能を持ちます。

5.3.25 n_unbind

(1) 機能概要

ネットワーク端点とネットワークアドレスとの結合を解放します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = n_unbind(fd);
int rcode, fd;
```

(3) 設定情報

(a) fd

ネットワーク端点を指定します。

(4) リターン情報

(a) リターンコード

0：正常終了

-1：異常終了

エラー番号が n_errno に設定されています。エラー番号の詳細は，「9.1 エラー番号一覧」を参照してください。

(5) 特記事項

この関数を発行したあとは，このネットワーク端点に対して，イベントやデータを受け付けません。

6 マクロ

この章では，マクロの文法について説明します。

6.1 リトライチェックマクロ

(1) 機能概要

reason で指定した理由コードをチェックして，リトライ情報を返します。

(2) 構文

```
#include <xnfw/niuser.h>
rcode = N_RETRYCK(reason);
int rcode;
unsigned int reason;
```

(3) 設定情報

(a) reason

n_rcvdis で取得した reason，または n_bind がエラーリターンした場合の，外部変数 n_comerr を指定します。

(4) リターン情報

(a) リターンコード

正数：正常終了（リトライ情報）

-1：異常終了

(5) 特記事項

リトライ情報を次に示します。

シンボリック名称	値（16 進）	再試行条件	説明
N_EROLDF	00000000	AP の独自基準で判定します。	相手ネットワークを解放します。
N_ERBUSY	00200000	一定時間間隔で複数回の再試行が望ましいです。	通信網の通信リソースが短期的にビジー状態です。
N_ERPDST	00400000		プロトコルによって，相手局の要因で異常終了しました。
N_ERNRTY	00600000		再試行が望ましい，そのほかの要因があります。
N_ERHARD	00800000	再試行は無意味です。	パラメタ不正です。
N_ERDEST	00a00000	再試行には，オペレータの判断が必要です。	通信網の通信リソースが長期的にビジー状態です。
N_ERPSRC	00c00000		自局のプロトコル上の要因で接続できません。

シンボリック名称	値（16 進）	再試行条件	説明
N_ERHALT	00e00000		オペレータの判断が必要な, そのほかの要因があります。

6.2 INS サービス種別設定マクロ

(1) 機能概要

従量課金（INS-P）/ 時間課金（INS-C）の種別，および INS-C の回線速度を設定します。このマクロ関数で返された値を，コネクション確立要求（n_connect）発行時に，アドレスフィールドの INS サービス種別に設定します。なお，VC 接続の ISDN 直結のときだけ有効となります。

このシステムは，ISDN 直結接続をサポートしていません。

(2) 構文

```
#include <xnfw/niuser.h>
ins_kind = X25_SET_INS(INS-P/C種別,INS-C回線速度);
unsigned short ins_kind;
```

(3) 設定情報

(a) INS-P/C 種別

INS-P/C 種別を指定します。内容を次に示します。

シンボリック名称	値（16 進）	意味
X25_INS_C	2000	INS-C です。
X25_INS_P	4000	INS-P です。
X25_INS_PC	8000	構成定義に従います。

(b) INS-C 回線速度

INS-C 回線速度を指定します。内容を次に示します。なお，INS-P の場合は，デフォルト値を指定してください。

シンボリック名称	値（16 進）	意味
X25_SPDDEF	0000	デフォルト値
X25_SPD02400	0008	2,400 bit/sec
X25_SPD04800	0009	4,800 bit/ sec
X25_SPD09600	000a	9,600 bit/sec
X25_SPD19200	000b	19,200 bit/sec
X25_SPD48000	000c	48,000 bit/sec
X25_SPD64000	000d	64,000 bit/sec

INS-C 回線速度に「X25_SPDDEF」を指定した場合は，構成定義で指定した INS-C 回線速度を使用します。

（4） リターン情報

（a） ins_kind

VC（ISDN 直結）接続時，コネクション確立要求（n_connect）の sndcall に指定する INS サービス種別です。設定値の詳細については，「7.1　ネットワークアドレス」を参照してください。

6.3 INS サービス種別取得マクロ

(1) 機能概要

従量課金（INS-P）/ 時間課金（INS-C）の種別を取得します。

コネクション確立指示，またはコネクション確立応答を受信した場合，n_call 構造体に INS サービス種別情報があるときは，このマクロで INS-P/C の種別を認識できます。

このシステムは，ISDN 直結接続をサポートしていません。

(2) 構文

```
#include <xnfw/niuser.h>
pc_value = X25_GET_PC(ins_kind);
unsigned short ins_kind;
unsigned short pc_value;
```

(3) 設定情報

(a) ins_kind

n_call 構造体に設定された INS サービス種別の値です。

設定値の詳細については，「7.1　ネットワークアドレス」を参照してください。

(4) リターン情報

(a) pc_value

確立されたコネクションの INS-P/C の種別です。設定値の詳細については，「6.2　INS サービス種別設定マクロ」を参照してください。

6.4 INS-C 回線速度取得マクロ

(1) 機能概要

INS-C の回線速度を取得します。

コネクション確立指示，またはコネクション確立応答を受信した場合，n_call 構造体に INS サービス種別情報があるときは，このマクロで INS-C の回線速度を認識できます。

このシステムは，ISDN 直結接続をサポートしていません。

(2) 構文

```
#include <xnfw/niuser.h>
spd_value = X25_GET_SPD(ins_kind);
unsigned short spd_value;
unsigned short ins_kind;
```

(3) 設定情報

(a) ins_kind

n_call 構造体に設定された INS サービス種別の値です。

設定値の詳細については，「7.1　ネットワークアドレス」を参照してください。

(4) リターン情報

(a) spd_value

INS-C で確立されたコネクションの回線速度です。設定値の詳細については，「6.2 INS サービス種別設定マクロ」を参照してください。

7 ネットワークアドレス仕様

この章では，ネットワークアドレスについて説明します。

7.1 ネットワークアドレス

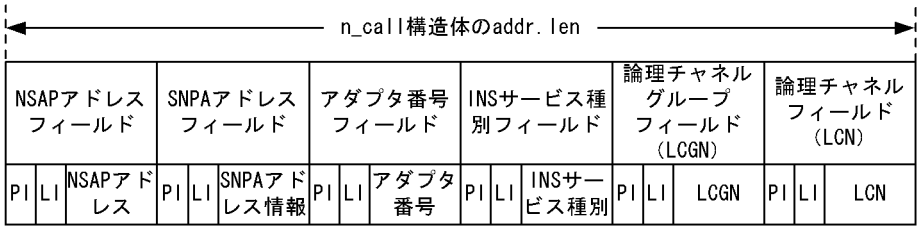
7.1 ネットワークアドレス

n_call 構造体には，次の情報を使用します。

- 相手局 NSAP アドレス
- 相手局 SNPA アドレス
- 自局アダプタ番号
- INS サービス種別
- 論理チャネルグループ番号
- 論理チャネル番号

VC 接続でのアドレスの形式を図 7-1 に，フィールドの内容を表 7-1 に示します。また，PVC 接続でのアドレスの形式を図 7-2 に，フィールドの内容を表 7-2 に示します。

図 7-1 アドレスの形式（VC 接続）



注

NSAP アドレスは，NSAP アドレス付加機能を使用している場合だけ有効です。SNPA アドレス情報とアダプタ番号は必須です。INS サービス種別は任意です。論理チャネル情報は，論理チャネル情報通知機能を使用している場合だけ，n_listen，n_rcvconnect，または同期の n_connect リターン時に設定されます。また，一般専用回線の場合，SNPA アドレス情報は通知されません。

表 7-1 フィールドの内容（VC 接続）

フィールド名称	長さ（単位：バイト）	値（16 進）	意味
PI	1	84	NSAP アドレス識別子
LI	1	00 ～ 14	NSAP アドレスの長さ
NSAP アドレス	0 ～ 20	2 進構文	NSAP アドレス
PI	1	D0	SNPA アドレス識別子
LI	1	02 ～ 10	SNPA アドレスの長さ

フィールド名称	長さ（単位：バイト）	値（16 進）	意味
SNPA アドレス	2 ～ 16	任意	SNPA アドレス 有効けた数（1バイト） | SNPAアドレス 左詰めで，奇数けたの場合は最終セミオクテッドを無視します。
PI	1	D2	アダプタ番号識別子
LI	1	4	アダプタ番号の長さ
アダプタ番号	4	任意	アダプタ番号
PI	1	D3	INS サービス種別識別子
LI	1	0 または 2	INS サービス種別の長さ
INS サービス種別	0 または 2	任意	INS ネットを使用する際の，従量課金（INS-P）/ 時間課金（INS-C）の種別，および INS-C の回線速度。 このシステムでは，ISDN 直結接続をサポートしていません。
PI	1	DE	論理チャネルグループ番号識別子
LI	1	1	論理チャネルグループ番号の長さ
論理チャネルグループ番号	1	任意	論理チャネルグループ番号
PI	1	DF	論理チャネル番号識別子
LI	1	1	論理チャネル番号の長さ
論理チャネル番号	1	任意	論理チャネル番号

注

PI がない場合，および LI が 0 の場合は，情報がないことを意味します。また，表 7-1 に示す PI 以外は無視されます。

図 7-2　アドレスの形式（PVC 接続）

注

アダプタ番号，論理チャネルグループ番号，および論理チャネル番号の各フィール

ドは必須です。

表 7-2 フィールドの内容（PVC 接続）

フィールド名称	長さ（単位：バイト）	値（16 進）	意味
PI	1	D2	アダプタ番号識別子
LI	1	4	アダプタ番号の長さ
アダプタ番号	4	任意	アダプタ番号
PI	1	D3	論理チャネルグループ番号識別子
LI	1	1	論理チャネルグループ番号の長さ
論理チャネルグループ番号	1	任意	論理チャネルグループ番号
PI	1	D4	論理チャネル番号識別子
LI	1	1	論理チャネル番号の長さ
論理チャネル番号	1	任意	論理チャネル番号
PI	1	D5	送信パケットサイズ識別子
LI	1	任意	送信パケットサイズの長さ
送信パケットサイズ	0 または 2	表 7-3 参照	送信パケットサイズ デフォルト値：128 送信パケットサイズの指定は，契約書に従って設定してください。
PI	1	D6	受信パケットサイズ識別子
LI	1	0 または 2	受信パケットサイズの長さ
受信パケットサイズ	0 または 2	表 7-3 参照	受信パケットサイズ デフォルト値：128 受信パケットサイズの指定は，契約書に従って設定してください。
PI	1	D7	送信ウィンドウサイズ識別子
LI	1	0 または 2	送信ウィンドウサイズの長さ
送信ウィンドウサイズ	0 または 1	1 ～ 7※	送信ウィンドウサイズ デフォルト値：2 送信ウィンドウサイズの指定は，契約書に従って設定してください。
PI	1	D8	受信ウィンドウサイズ識別子
LI	1	0 または 1	受信ウィンドウサイズの長さ
受信ウィンドウサイズ	0 または 1	1 ～ 7※	受信ウィンドウサイズ デフォルト値：2 受信ウィンドウサイズの指定は，契約書に従って設定してください。

注※

1 ～ 7 以外を指定した場合は 2 が仮定されます。

表 7-3　パケットサイズの指定内容

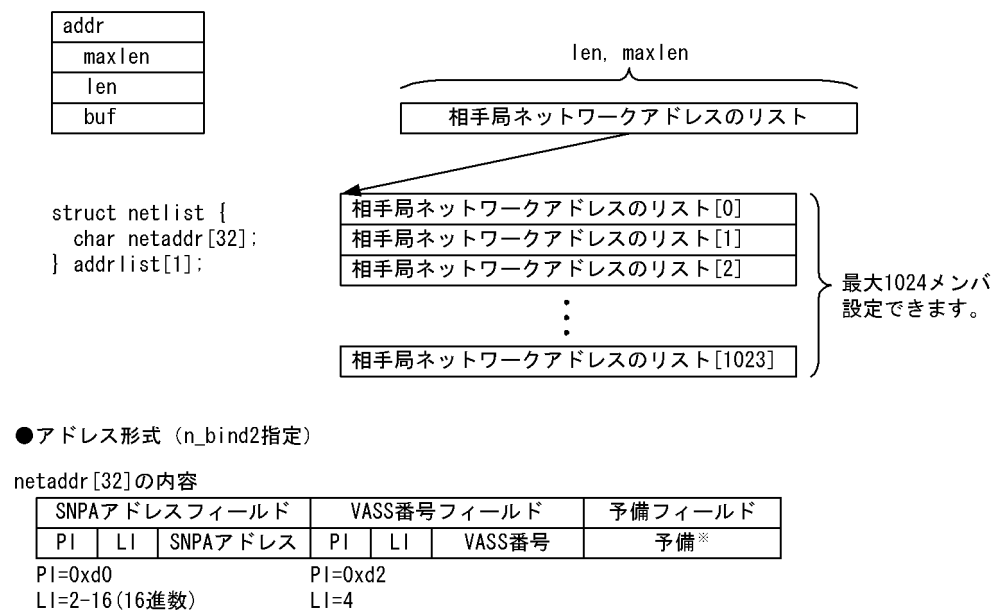
値	意味
0x0080	128 バイト
0x0100	256 バイト
0x0200	512 バイト
0x0400	1,024 バイト
0x0800	2,048 バイト
0x1000	4,096 バイト

n_bind2，n_chglist の n_bind 構造体には，次の情報を設定します。

- 相手局 SNPA アドレス
- 自局アダプタ番号

addr で指定する相手局ネットワークアドレスを図 7-3 に示します。

図 7-3　addr で指定する相手局ネットワークアドレス



注 1

SNPA アドレスを指定し，VASS 情報に関する情報を省略した場合，該当する SNPA アドレスの全 VASS 番号が対象となります。

注 2

VASS 番号を指定し，SNPA アドレス情報に関する情報を省略した場合，該当する VASS 番号の全 SNPA アドレスが対象となります。

注 3

SNPA アドレス情報と VASS 番号の両方を指定していない場合，n_errno=NBADADDR でエラーとなります。

注 4

指定した相手局ネットワークアドレスが，ほかの端点で指定されていてもエラーとしません。

注※

予備フィールドには，0 を埋めてください。

8 ネットワークプロトコルオプション

この章では，ネットワークプロトコルオプションについて説明します。

8.1 サポートするオプション

8.1 サポートするオプション

サポートするプロトコルオプションを次に示します。

- パケットサイズ
- 優先データ使用有無

オプション形式を図 8-1 に，フィールドの内容を表 8-1 に示します。なお，優先データ使用有無フィールドについては，「11.1 優先データ受信機能」を参照してください。

図 8-1 オプション形式

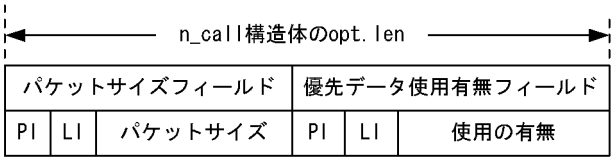


表 8-1 フィールドの内容

フィールド名称	長さ（単位：バイト）	値（16 進）	意味
PI	1	85	パケットサイズの識別子
LI	1	0 または 2	パケットサイズの長さ
パケットサイズ	0 または 2	0x0080	128 バイト
		0x0100	256 バイト
		0x0200	512 バイト
		0x0400	1,024 バイト
		0x0800	2,048 バイト
		0x1000	4,096 バイト
PI	1	8A	優先データ使用の有無の識別子
LI	1	0 または 2	使用の有無の長さ
優先データ使用の有無	0 または 2	0x0000	使用しません
		0x0001	使用します

9 エラーコード

この章では，提供関数が異常終了した場合のエラーコードについて説明します。

9.1 エラー番号一覧

提供関数が異常終了した場合，n_errno にエラー番号が設定されます。エラー番号一覧を表 9-1 に示します。また，各提供関数が設定するエラー番号の一覧を表 9-2 〜表 9-4 に示します。

表 9-1 エラー番号一覧

エラー番号	意味
NBADADDR	指定したプロトコルアドレスの形式が誤っています。
NBADDATA	ユーザデータの量が，ネットワークプロバイダが許可する限界を超えました。
NBADF	指定したファイル記述子が示すネットワーク端点が不正です。
NBADFLAG	無効なフラグを指定しました。
NBADOPT	指定したプロトコルオプションの形式が誤っています。
NBADSEQ	無効なシーケンス番号を指定しました。
NBUFOVFLW	入ってくる引数値に対して，許可しているバイト数がその引数値を収めるのに十分な値ではありません。
NCOMERR	ネットワークプロバイダに対する要求が異常終了しました。このとき，n_comerr にはその原因を示すコードが設定されます。
NEPARAM	指定したパラメタが不正です。
NFLOW	ネットワークプロバイダがフロー制御中，または内部資源解放待ちであるが，非同期です。
NLOOK	該当するネットワーク端点で非同期のイベントが発生しているため，それに対してすぐに対応する必要があります。
NNODATA	非同期であり，現在処理できるデータがありません。
NNODIS	該当するネットワーク端点には，その時点で接続解除の指示がありません。
NNOERR	該当するネットワーク端点には，エラーが発生していません。
NNOTSUPPORT	この関数はサポートされていません。
NOUTSTATE	関数の指定順序が正しくありません。
NOVRCON	接続指示数が，ネットワークプロバイダの限界を超えました。
NSTATECHNG	ネットワークプロバイダが，状態を変えようとしています。
NSYSERR	システムエラーが発生しました。グローバル変数 errno を参照してください。XNF/AS/NLI が設定する詳細エラー番号については，「表 9-5　NLI が設定する詳細エラー番号」を参照してください。

表 9-2　各提供関数が設定するエラー番号の一覧（1/3）

エラー番号	関数名称								
	n_accept	n_alloc	n_bind	n_close	n_connect	n_error	n_free	n_getinfo	n_rstrsp
NBADADDR	○		○		○				
NBADDATA									
NBADF	○	○	○	○	○		○	○	○
NBADFLAG									
NBADOPT	○				○				
NBADSEQ	○								
NBUFOVFLW	○		○		○※1				
NCOMERR			○						
NEPARAM	○	○	○	○	○	○		○	
NFLOW									
NLOOK	○				○				
NNODATA					○※2				
NNODIS									
NNOERR						○			
NNOTSUPPORT									○
NOUTSTATE	○		○		○				○
NOVRCON					○				
NSTATECHNG	○		○		○※1			○	
NSYSERR	○	○	○	○	○	○		○	○

（凡例）

○：該当するエラー番号を設定します。

空白：エラー番号を設定しません。

注※ 1

同期の場合に設定します。

注※ 2

非同期の場合に設定します。

表 9-3 各提供関数が設定するエラー番号の一覧（2/3）

エラー番号	関数名称							
	n_getstate	n_listen	n_look	n_open	n_rcv	n_rcvconnect	n_rcvdis	n_snd
NBADADDR						○		
NBADDATA								
NBADF	○	○	○		○	○	○	○
NBADFLAG								
NBADOPT								
NBADSEQ								
NBUFOVFLW		○※1				○		
NCOMERR								
NEPARAM	○	○	○	○	○	○	○	○
NFLOW								○※2
NLOOK					○	○	○	○
NNODATA		○※2			○※2	○		
NNODIS							○	
NNOERR								
NNOTSUPPORT								
NOUTSTATE		○			○	○	○	○
NOVRCON								
NSTATECHNG			○		○			○※1
NSYSERR	○	○	○	○	○	○	○	○

（凡例）

○：該当するエラー番号を設定します。

空白：エラー番号を設定しません。

注※ 1

同期の場合に設定します。

注※ 2

非同期の場合に設定します。

表 9-4　各提供関数が設定するエラー番号の一覧（3/3）

エラー番号	関数名称							
	n_snddis	n_sync	n_unbind	n_sndrst	n_rcvrst	n_rcvint	n_bind2	n_chglist
NBADADDR							○	○
NBADDATA								
NBADF	○	○	○	○	○	○	○	○
NBADFLAG								
NBADOPT								
NBADSEQ	○							
NBUFOVFLW								
NCOMERR								
NEPARAM	○	○	○	○	○	○	○	○
NFLOW								
NLOOK	○			○		○		
NNODATA						○		
NNODIS								
NNOERR								
NNOTSUPPORT					○	○	○	○
NOUTSTATE	○		○	○	○	○	○	○
NOVRCON								
NSTATECHNG	○	○						
NSYSERR	○	○	○	○	○	○	○	○

（凡例）

○：該当するエラー番号を設定します。

空白：エラー番号を設定しません。

9.2 詳細エラー番号

提供関数が異常終了して，n_errno に NSYSERR が設定されている場合，グローバル変数 errno に詳細エラー番号が設定されます。

プログラム中で詳細エラー番号を参照する場合は，シンボリック名を使用します。なお，グローバル変数 errno は，提供関数が正常終了した場合はクリアされません。

NLI が設定する詳細エラー番号を表 9-5 に示します。そのほかの詳細エラー番号については，AIX マニュアルを参照してください。

表 9-5 NLI が設定する詳細エラー番号

シンボリック名	値	意味	ユーザの処置
EINTR	4	シグナルを受信しました。	アプリケーションプロセスを終了してください。
ENXIO	6	通信管理が動作できなくなりました。	アプリケーションプロセスを終了して，通信管理を起動してください。
EFAULT	14	不正な領域が指定されているため，アクセス時にメモリフォルトが発生しました。	正しいアドレスを指定して，再試行してください。
EBUSY	16	リソースビジー状態です。	再試行してください。

9.3 切断理由コード

切断理由コードは，コネクション解放時に設定されるコードです。

NLI が生成する切断理由コードを表 9-6 に示します。そのほかのコードについては，マニュアル「通信管理　XNF/AS　解説・運用編」を参照してください。

表 9-6　NLI が生成する切断理由コード

コード	意味
0x034100f4	VC 接続サーバ端点がないため，NLI で拒否しました。
0x03000000	VC 接続 NLI のエンドユーザの要求で解放・拒否しました。
0x046f0001	PVC 接続 xnfoffline コマンドで，回線がオフラインにされました。

10 プログラムの作成

この章では，プログラムの作成，およびインタフェース構造体のパラメタについて説明します。

10.1 ヘッダファイル

NLI 通信機能を使用する上位プログラムは，表 10-1 のヘッダファイルを取り込む必要があります。

表 10-1 NLI に関係するヘッダファイル

ファイル名	書き方	説明
niuser.h	#include <xnfw/niuser.h>	NLI 通信機能で使用する構造体およびマクロの定義情報です。
errno.h	#include <errno.h>	グローバル変数 errno を参照する際に使用します。

10.2　ライブラリ

NLI 通信機能を使用する上位プログラムは，リンケージ実行時に NLI 通信機能用のライブラリ関数のオブジェクトモジュールを取り込む必要があります。このため，コンパイルとリンクのときには，ライブラリを指定する必要があります。共用ライブラリを使用することによって，NLI 通信機能のバージョンアップでライブラリが変更されても，上位プログラムを再リンケージする必要がなくなります。NLI 通信機能が提供するライブラリファイルを表 10-2 に，NLI 通信機能のライブラリファイルとのリンクを表 10-3 に示します。

表 10-2　NLI 通信機能が提供するライブラリファイル

ライブラリ種別	説明
共用ライブラリ（/lib/libNSP.so）	ライブラリ関数は，実行時に上位プログラムと結合されます。
共用ライブラリ（/lib/libNSPex1.so）	ライブラリ関数は，実行時に上位プログラムと結合されます。「優先データ受信機能」，「上位プロセス選択機能」を使用する場合に，既存のライブラリとあわせて使用します。
共用ライブラリ（/lib/libNSPex2.so）	ライブラリ関数は，実行時に上位プログラムと結合されます。「RF パケット応答同期機能」を使用する場合に，既存のライブラリとあわせて使用します。

表 10-3　NLI 通信機能のライブラリファイルとのリンク

リンク対象ライブラリ種別	リンク方法
共用ライブラリ（/lib/libNSP.so）	xlc ファイル名 -brtl -lNSP
共用ライブラリ（/lib/libNSPex1.so）	xlc ファイル名 -brtl -lNSP -lNSPex1
共用ライブラリ（/lib/libNSPex2.so）	xlc ファイル名 -brtl -lNSP -lNSPex1 -lNSPex2

10.3 インタフェース構造体のパラメタ

各インタフェース構造体のパラメタの設定・参照を表 10-4 ～表 10-11 に示します。

表 10-4 インタフェース構造体（n_info）のパラメタの設定・参照

構造体名称	パラメタ名称またはメンバ名	関数名	
		n_getinfo	n_open
n_info	info	○	○
	addr	△	△
	options	△	△
	nsdu	△	△
	ensdu	△	△
	connect	△	△
	discon	△	△
	servtype	△	△

（凡例）

○：設定できます。

△：参照できます。

表 10-5 インタフェース構造体（n_bind）のパラメタの設定・参照（その 1）

構造体名称	パラメタ名称またはメンバ名	関数名	
		n_bind(VC)	n_bind(PVC)
n_bind	req	○	◎
	addr	－	◎
	maxlen	－	◎
	len	－	◎
	buf	－	◎
	qlen	○	－
	ret	○	○
	addr	－	△
	maxlen	－	△
	len	－	△
	buf	－	△
	qlen	△	－

（凡例）

◎：設定する必要があります。
○：設定できます。
△：参照できます。
－：設定，参照する必要はありません。

表 10-6　インタフェース構造体（n_bind）のパラメタの設定・参照（その 2）

構造体名称	パラメタ名称またはメンバ名	関数名
		n_bind2(VC)
n_bind	req	○
	addr	○
	maxlen	○
	len	○
	buf	○
	qlen	○
	ret	○
	addr	－
	maxlen	－
	len	－
	buf	－
	qlen	△

（凡例）

○：設定できます。
△：参照できます。
－：設定，参照する必要はありません。

表 10-7　インタフェース構造体（n_bind）のパラメタの設定・参照（その 3）

構造体名称	パラメタ名称またはメンバ名	関数名
		n_chglist(VC)
n_bind	chgbind	○
	addr	○
	maxlen	○
	len	○
	buf	○
	qlen	－

（凡例）
○：設定できます。
－：設定，参照する必要はありません。

表 10-8 インタフェース構造体（n_discon）のパラメタの設定・参照

構造体名称	パラメタ名称またはメンバ名	関数名
		n_rcvdis
n_discon	discon	○
	udata	○
	maxlen	○
	len	△
	buf	△
	reason	△
	sequence	△

（凡例）
○：設定できます。
△：参照できます。

表 10-9 インタフェース構造体（n_reset）のパラメタの設定・参照

構造体名称	パラメタ名称またはメンバ名	関数名	
		n_sndrst	n_rcvrst
n_reset	req	◎	－
	ver	－	－
	cause	◎	－
	diag	◎	－
	cdgflg	－	－
	ret	○	－
	ver	－	－
	cause	△	－
	diag	△	－
	cdgflg	△	－
	reset	－	○
	ver	－	－
	cause	－	△

構造体名称	パラメタ名称またはメンバ名	関数名	
		n_sndrst	n_rcvrst
	diag	―	△
	cdgflg	―	△

（凡例）

◎：設定する必要があります。
○：設定できます。
△：参照できます。
―：設定，参照する必要はありません。

表 10-10　インタフェース構造体（n_call）のパラメタの設定・参照（その 1）

構造体名称	パラメタ名称またはメンバ名	関数名			
		n_accept	n_listen	n_rcvconnect	n_snddis
n_call	call	◎※2	◎※2	○※2	◎※2
	addr	○	○	○	―
	maxlen	○	○	○	―
	len	○	△	△	―
	buf	○	△	△	―
	opt	○	○	○	―
	maxlen	○	○	○	―
	len	○	△	△	―
	buf	○	△	△	―
	udata	○	○	○	○
	maxlen	○	○	○	○
	len	○	△	△	○
	buf	○	△	△	○
	sequence	◎	△※2	―	◎※1

（凡例）

◎：設定する必要があります。
○：設定できます。
△：参照できます。
―：設定，参照する必要はありません。

注※ 1

コネクション確立を拒否する場合は必要です。

注※ 2

PVC では無視されます。NULL を指定しても問題ありません。

表 10-11 インタフェース構造体（n_call）のパラメタの設定・参照（その 2）

構造体名称	パラメタ名称またはメンバ名	関数名	
		n_connect（同期）	n_connect（非同期）
n_call	sndcall	◎※	◎※
	addr	◎	◎
	maxlen	◎	◎
	len	◎	◎
	buf	◎	◎
	opt	○	○
	maxlen	○	○
	len	○	○
	buf	○	○
	udata	○	○
	maxlen	○	○
	len	○	○
	buf	○	○
	sequence	−	−
	rcvcall	○	−
	addr	○	−
	maxlen	○	−
	len	△	−
	buf	△	−
	opt	○	−
	maxlen	○	−
	len	△	−
	buf	△	−
	udata	○	−
	maxlen	○	−
	len	△	−

構造体名称	パラメタ名称またはメンバ名	関数名	
		n_connect（同期）	n_connect（非同期）
	buf	△	－
	sequence	－	－

（凡例）

◎：設定する必要があります。
○：設定できます。
△：参照できます。
－：設定，参照する必要はありません。

注※

PVC では無視されます。NULL を指定しても問題ありません。

11 付加機能

この章では，NLI の付加機能について説明します。

11.1 優先データ受信機能

(1) 優先データ（IT パケット）サービスサポートの有無指定

n_open 発行時，oflag に O_XNFNLI_INT を指定することで，優先データが使用できるようになります。なお，n_open の oflag に O_XNFNLI_INT を指定していない場合，優先データ使用有無の opt フィールドは設定されません。

(2) コネクション確立時の使用有無の参照

コネクション確立時の，各関数での優先データ使用有無の opt フィールドの設定・参照を表 11-1 に，自局と相手局のネゴシエーション関連を表 11-2 に示します。

表 11-1 優先データ使用有無の opt フィールドの設定・参照

エリア名称	フィールド	関数名				
		n_connect（同期）	n_connect（非同期）	n_accept	n_listen	n_rcvconnect
sndcall	opt	―	―	―	―	―
rcvcall	opt	○	―	―	―	―
call	opt	―	―	―	○	○

（凡例）
○：参照できます。
―：設定・参照する必要はありません。

表 11-2 自局と相手局のネゴシエーション関連

相手局	自局	
	n_open の oflag で O_XNFNLI_INT 指定なし	n_open の oflag で O_XNFNLI_INT 指定あり
コネクション確立指示 / 確認で優先データ使用なし	×	×
コネクション確立指示 / 確認で優先データ使用あり	×	○

（凡例）
○：優先データ使用ありです。
×：優先データ使用なしです。

(3) 優先データ受信イベント通知

優先データを受信すると，n_look に対して N_INT が応答されます。

(4) 優先データのユーザデータ受信

優先データを受信したとき，n_rcvint を発行することによって優先データのユーザデータ（最大 32 バイト）を受信できます。なお，n_rcvint の受信バッファサイズが受信データ長より小さい場合，受信バッファサイズ分を設定して，残りのデータは次の n_rcvint で読み込みます。ただし，分割に関する情報は設定されません。

11.2 コールユーザデータの転送機能

(1) コールユーザデータサービスサポートの有無

n_open 発行時，oflag に O_XNFNLI_CALL を指定することで，コールユーザデータの転送機能が使用できるようになります。

(2) コールユーザデータの送受信

コネクション確立時の各関数でのコールユーザデータの設定・参照を表 11-3 に示します。

表 11-3 コールユーザデータの設定・参照

エリア名称	フィールド	関数名				
		n_connect（同期）	n_connect（非同期）	n_accept	n_listen	n_rcvconnect
sndcall	udata	○	○	－	－	－
rcvcall	udata	△	－	－	－	－
call	udata	－	－	○	△	△

（凡例）
○：設定できます。
△：参照できます。
－：設定・参照する必要はありません。

注意事項

1. この機能を使用する場合，同一回線での OSI との共存はできません。また，構成定義では，使用する回線に属する NL または X25_group_define 文の専用回線識別子を，X.25 パススルー専用回線にする必要があります。
2. ユーザが送信要求したコールユーザデータが最大長を超えた場合，最大長まで付加して送信します（最大長については，n_open のリターン値を参照）。
3. ユーザが指定した udata フィールドの最大長が，受信したコールユーザデータを超えた場合，超過分を破棄します。

11.3　Q ビットの付加および通知機能

（1）Q ビットサポートの有無

n_open 発行時，oflag に O_XNFNLI_QBIT を指定することで，Q ビットの付加・通知機能が使用できるようになります。なお，n_open での指定は VC 接続のときだけ必要であり，PVC 接続で Q ビットを使用する場合は n_open での指定は必要ありません。

（2）Q ビットの付加

n_snd 発行時，flags に論理和形式で N_QBIT を設定することで，回線上のパケットに Q ビットを付加します。

（3）Q ビットの通知

受信パケットに Q ビットが付加されている場合，該当するデータを受信する n_rcv の flags の N_QBIT を ON にして，AP に通知します。

11.4 論理チャネルグループ番号，論理チャネル番号の通知機能

(1) 論理チャネル情報（論理チャネルグループ番号，論理チャネル番号）の通知

n_open 発行時，oflag に O_XNFNLI_LCN を指定することで，論理チャネル情報の通知機能が使用できるようになります。

(2) 論理チャネル情報の通知

コネクション確立時の各関数での論理チャネル情報の参照を表 11-4 に示します。

表 11-4 論理チャネル情報の参照

エリア名称	フィールド	関数名				
		n_connect（同期）	n_connect（非同期）	n_accept	n_listen	n_rcvconnect
sndcall	addr	—	—	—	—	—
rcvcall	addr	○	—	—	—	—
call	addr	—	—	—	○	○

（凡例）

○：参照できます。

—：設定・参照する必要はありません。

11.5 非同期切断機能

(1) 非同期切断機能の有無

n_open 発行時，oflag に O_XNFNLI_DISC を指定することで，非同期切断機能が使用できるようになります。

(2) 非同期切断

n_snddis 発行時，非同期切断機能を使用していると，相手局の応答を待たないで関数がリターンします。

注意事項

非同期切断インタフェースを使用する場合，下位層でコネクション解放に時間を要すると，その後確立できるコネクション数が一時的に少なくなることがあります。

11.6 上位プロセス選択機能

(1) 上位プロセス選択機能の有無

n_open 発行時，oflag に O_XNFNLI_MLT を指定することで，上位プロセス選択機能が使用できるようになります。

(2) 上位プロセス選択機能

n_bind2 発行時に指定した相手局ネットワークアドレスで，コネクション確立指示の振り分けを行います。また，n_bind2 発行時に指定した相手局ネットワークアドレスの変更もできます。

注意事項

1. コネクション確立指示受信時，n_bind と n_bind2 の両方を発行している場合，n_bind2 を優先して通知します。
2. 同じ相手局ネットワークアドレスを指定した n_bind2 が複数発行している場合，どのネットワーク端点に通知するか保証できません。

11.7 NSAP アドレス付加機能

(1) NSAP アドレス通知機能の有無

n_open 発行時，oflag に O_XNFNLI_NSAP を指定することで，NSAP アドレス付加機能が使用できるようになります。

(2) NSAP アドレス送信

n_connect 発行時，NSAP アドレス付加機能を使用して，sndcall の addr フィールドに相手局 NSAP アドレスを指定すると，CR パケットに NSAP アドレスを付加して送信します。

(3) NSAP アドレス受信

n_listen 発行時，NSAP アドレス付加機能を使用して，CN パケットに相手局 NSAP アドレスが付加されている場合，NSAP アドレスを call の addr フィールドに設定してユーザに通知します。

注意事項

NSAP アドレス付加機能を使用する場合，84VC だけ使用できます。また，同一回線で OSI との共存はできません。構成定義では，使用する回線に属する NL または X25_group_define 文の専用回線識別子を，X.25 パススルー専用回線にする必要があります。

11.8 RF パケット応答同期機能

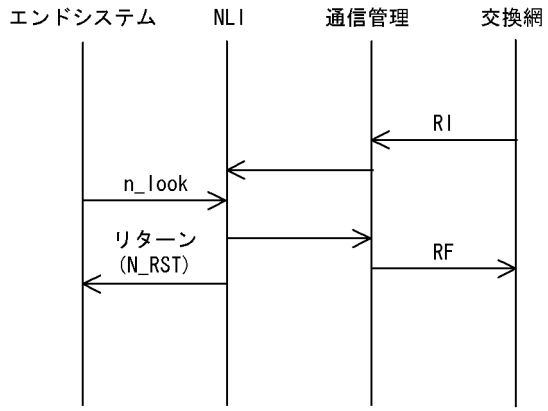
(1) RF パケット応答同期機能の有無

n_open 発行時，oflag に O_XNFNLI_RSTRSP を指定することで，RF パケット応答同期機が使用できるようになります。

(2) RF パケット応答同期機能

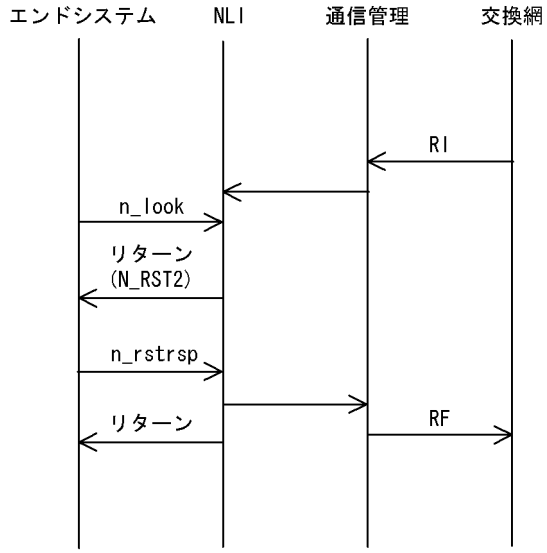
RF パケット応答同期機能を使用しない場合，ネットワークコネクションのリセット指示をすると，図 11-1 に示すように n_look 関数に対して N_RST イベントを通知し，RF パケットを応答します。

図 11-1 ネットワークコネクションのリセット手順（RF パケット応答同期機能を使用しない場合）



RF パケット応答同期機能を使用した場合，ネットワークコネクションのリセット指示を受信すると，図 11-2 に示すように n_look 関数に対して N_RST2 イベントを通知しますが，RF パケットは応答しません。n_rstrsp 関数を発行することで，上位プログラムから RF パケットを応答できます。

図 11-2 ネットワークコネクションのリセット手順（RF パケット応答同期機能を使用する場合）



RF パケット応答同期機能使用時のネットワークコネクションのリセットで使用する関数を表 11-5 に示します。

表 11-5 ネットワークコネクションのリセットで使用する関数（RF パケット応答同期機能使用時）

関数	説明
n_look	ネットワークコネクションのリセット指示を受信して，リセット指示（N_RST2 イベント）を通知します。RF パケットは応答しません。
n_rstrsp	RF パケットを応答します。

注意事項

- n_rstrsp 関数は，コネクションの解放事象を受信していると，RF パケットを送信しないで正常リターンします。
- n_look 関数で N_RST2 イベントを通知するとき，ほかのイベントと異なり再度 n_look 関数を発行してもイベントなしでリターンします。
- n_look 関数で N_RST2 イベントを通知したあと，n_rstrsp 関数の発行待ち状態のときに，n_close 関数または close 関数が発行されると，次のようになります。
  VC 接続の場合：RF パケットを送信しないでネットワークコネクションを解放します。
  PVC 接続の場合：RF パケットを送信します。

付録

付録 A　従来製品との差異

付録 A　従来製品との差異

従来製品（HI-UX/WE2 上で動作する XNF/S-E2）との差異について説明します。

(1) ライブラリ

ライブラリは，共用ライブラリだけ提供します。アーカイブライブラリは提供しません。ライブラリの使用方法については，「10.　プログラムの作成」を参照してください。

(2) 詳細エラーコード

通信管理がメモリにロードされていない場合，詳細エラーコードは ENODEV ではなく ENXIO になります。

(3) X25 シリーズマクロ

X25_SET_INS，X25_GET_PC，および X25_GET_SPD のマクロは，XNF/S-E2 との互換※のため，残しています。XNF/AS は ISDN 直結接続をサポートしないため，これらのマクロは使用できません。

注※

XNF/AS の動作は，XNF/S-E2 で ISDN 直結接続以外（例えば，専用回線接続や私設パケット交換接続）の構成接続をした場合と同じになります。

索引

S

U

あ

い

え

か

き

こ

さ

し

せ

ち

つ

て

と

ね

は

ひ

ふ

へ

ま

ゆ

ら

り

ろ

ソフトウェアマニュアルのサービス ご案内

１．マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しています。

URL　http://www.hitachi.co.jp/soft/manual/

ホームページのメニューは次のとおりです。

■マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ，マニュアル名称，資料番号のいずれかから検索できます。
■CD-ROMマニュアル	日立ソフトウェアマニュアルと製品群別CD-ROMマニュアルの仕様について記載しています。
■マニュアルのご購入	マニュアルご購入時のお申し込み方法を記載しています。
■オンラインマニュアル	一部製品のマニュアルをインターネットで公開しています。
■サポートサービス	ソフトウェアサポートサービスお客様向けページでのマニュアル公開サービスを記載しています。
■ご意見・お問い合わせ	マニュアルに関するご意見，ご要望をお寄せください。

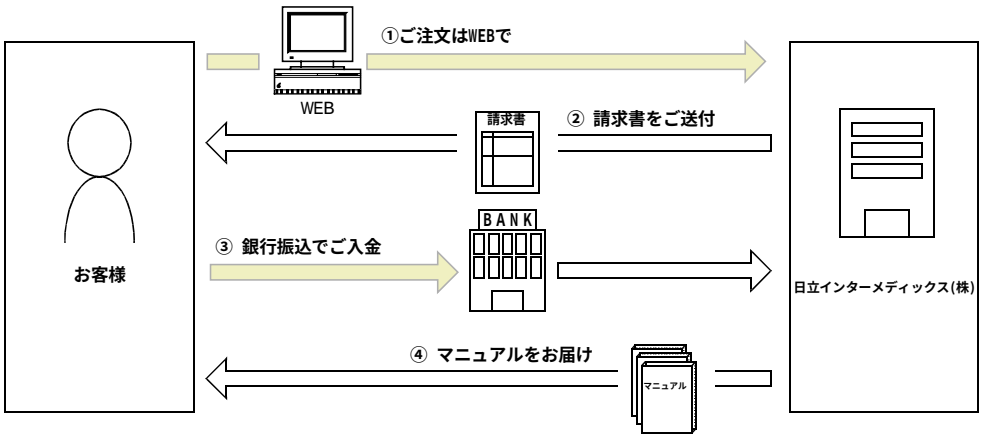
２．インターネットでのマニュアル公開

2 種類のマニュアル公開サービスを実施しています。

(1) マニュアル情報ホームページ「オンラインマニュアル」での公開
　製品をよりご理解いただくためのご参考として，一部製品のマニュアルを公開しています。

(2) ソフトウェアサポートサービスお客様向けページでのマニュアル公開
　ソフトウェアサポートサービスご契約のお客様向けにマニュアルを公開しています。公開しているマニュアルの一覧，本サービスの対象となる契約の種別などはマニュアル情報ホームページの「サポートサービス」をご参照ください。

３．マニュアルのご注文



① マニュアル情報ホームページの「マニュアルのご購入」にアクセスし，お申し込み方法をご確認のうえ WEB からご注文ください。ご注文先は日立インターメディックス(株)となります。

② ご注文いただいたマニュアルについて請求書をお送りします。

③ 請求書の金額を指定銀行へ振り込んでください。

④ 入金確認後 7 日以内にお届けします。在庫切れの場合は，納期を別途ご案内いたします。