Oracle® Solaris 管理: ネットワークインタフェースとネットワーク仮想化
このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、翻訳、改変、転送、ライセンス供与、送信、配布、発表、施行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブリ、逆コンパイルは保険のためには法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オアクラル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007).

Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このソフトウェアもしくはハードウェアは様々な情報管理アプライケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプライケーション（人への傷害を発生させる可能性があるアプライケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアは危険が伴うアプライケーションで使用する際、安全に使用するために、適切な安全対策、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプライケーションで使用したことにより因って損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel、Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD、OpteronおよびAMD OpteronはAdvanced Micro Devices, Inc.の商標または登録商標です。

UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報提供を目的があります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。
目次

はじめに ........................................................................................................... 15

1 ネットワークスタックの概要 ............................................................................ 21
この Oracle Solaris リリースでのネットワーク構成 ........................................ 21
Oracle Solaris のネットワークスタック .......................................................... 22
ネットワークデバイスとデータリンク名 ............................................................. 26
  デフォルトの汎用リンク名 ........................................................................ 26
  データリンクへの汎用名の割り当て ................................................................. 27
  汎用リンク名の割り当て方法をカスタマイズする ........................................ 28
  アップグレードされたシステムでのリンク名 ................................................. 29
  その他のリンクタイプの管理 ..................................................................... 31

パート I Network Auto-Magic ............................................................................. 35

2 NWAM の紹介 .................................................................................................... 37
NWAM 構成とは .................................................................................................. 38
  NWAM の機能コンポーネント ................................................................ 39
NWAM を使用する場合 ....................................................................................... 40
NWAM 構成の動作 ............................................................................................. 41
  NWAM のデフォルト動作 ......................................................................... 42
  その他の Oracle Solaris ネットワーク技術を使用した NWAM の動作 .......... 42
ネットワーク構成タスクを検索する場所 ............................................................... 44

3 NWAM 構成と管理 (概要) ............................................................................ 47
NWAM 構成の概要 ............................................................................................. 47
  ネットワークプロファイルとは ................................................................ 47
NCP の説明 ....................................................................................................... 48
<table>
<thead>
<tr>
<th>テキスト</th>
<th>ページ</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCU の説明</td>
<td>49</td>
</tr>
<tr>
<td>Automatic NCP とユーザー定義の NCP の説明</td>
<td>50</td>
</tr>
<tr>
<td>場所プロファイルの説明</td>
<td>50</td>
</tr>
<tr>
<td>ENM の説明</td>
<td>51</td>
</tr>
<tr>
<td>既知のWLANについて</td>
<td>52</td>
</tr>
<tr>
<td>NWAM構成データ</td>
<td>53</td>
</tr>
<tr>
<td>NCUプロパティの値</td>
<td>54</td>
</tr>
<tr>
<td>システム定義の場所のプロパティ値</td>
<td>56</td>
</tr>
<tr>
<td>NWAMプロファイルをアクティブにする方法</td>
<td>58</td>
</tr>
<tr>
<td>NCPアクティブ化ポリシー</td>
<td>59</td>
</tr>
<tr>
<td>場所のアクティブ化の選択条件</td>
<td>61</td>
</tr>
<tr>
<td>netcfgを使用してプロファイルを構成する</td>
<td>63</td>
</tr>
<tr>
<td>netcfg対話型モード</td>
<td>65</td>
</tr>
<tr>
<td>netcfgコマンド行モード</td>
<td>65</td>
</tr>
<tr>
<td>netcfgコマンドファイルモード</td>
<td>66</td>
</tr>
<tr>
<td>netcfgでサポートされるサブコマンド</td>
<td>66</td>
</tr>
<tr>
<td>netadmコマンドを使用してプロファイルを管理する</td>
<td>69</td>
</tr>
<tr>
<td>NWAMデーモンの概要</td>
<td>71</td>
</tr>
<tr>
<td>NWAMポリシーエンジンデーモン(nwamd)の説明</td>
<td>71</td>
</tr>
<tr>
<td>NWAMリポジトリデーモン(netcfgd)の説明</td>
<td>72</td>
</tr>
<tr>
<td>SMFネットワークサービス</td>
<td>72</td>
</tr>
<tr>
<td>NWAMセキュリティーの概要</td>
<td>73</td>
</tr>
<tr>
<td>NWAMに関連する承認とプロファイル</td>
<td>74</td>
</tr>
<tr>
<td>NWAMユーザーメンタフェースを使用するために必要な承認</td>
<td>74</td>
</tr>
</tbody>
</table>

4 NWAM プロファイルの構成(タスク) ..................................................77

<table>
<thead>
<tr>
<th>テキスト</th>
<th>ページ</th>
</tr>
</thead>
<tbody>
<tr>
<td>プロファイルを作成する</td>
<td>78</td>
</tr>
<tr>
<td>プロファイルを作成</td>
<td>78</td>
</tr>
<tr>
<td>プロファイルを対話形式で作成する</td>
<td>79</td>
</tr>
<tr>
<td>NCPを作成する</td>
<td>80</td>
</tr>
<tr>
<td>NCPにNUCを作成する</td>
<td>80</td>
</tr>
<tr>
<td>▼ NCPを対話形式で作成する方法</td>
<td>83</td>
</tr>
<tr>
<td>場所プロファイルを作成する</td>
<td>88</td>
</tr>
<tr>
<td>ENMプロファイルを作成する</td>
<td>93</td>
</tr>
<tr>
<td>WLANを作成する</td>
<td>96</td>
</tr>
<tr>
<td>目次</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td></td>
</tr>
<tr>
<td>プロファイルを削除する ................................................................. 98</td>
<td></td>
</tr>
<tr>
<td>プロファイルのプロバティー値を設定および変更する .............................. 100</td>
<td></td>
</tr>
<tr>
<td>プロファイル情報についてシステムに問い合わせる ................................ 102</td>
<td></td>
</tr>
<tr>
<td>システム上のプロファイルをすべて一覧表示する .................................. 102</td>
<td></td>
</tr>
<tr>
<td>特定のプロファイルのプロバティー値をすべて一覧表示する ..................... 103</td>
<td></td>
</tr>
<tr>
<td>特定のプロバティー値を取得する .................................................. 104</td>
<td></td>
</tr>
<tr>
<td>walkpropサブコマンドを使用して、プロバティー値を対話形式で表示および変更する ........................................ 106</td>
<td></td>
</tr>
<tr>
<td>プロファイル構成をエクスポートおよび復元する ................................ 108</td>
<td></td>
</tr>
<tr>
<td>ユーザー定義のプロファイルを復元する ........................................ 111</td>
<td></td>
</tr>
<tr>
<td>ネットワーク構成の管理 ......................................................... 112</td>
<td></td>
</tr>
<tr>
<td>▼自動ネットワーク構成モードから手動ネットワーク構成モードに切り替える方法 ...................................... 112</td>
<td></td>
</tr>
<tr>
<td>▼手動ネットワーク構成モードから自動ネットワーク構成モードに切り替える方法 ...................................... 113</td>
<td></td>
</tr>
<tr>
<td>5 NWAM プロファイルの管理(タスク) ................................................. 115</td>
<td></td>
</tr>
<tr>
<td>プロファイルの状態に関する情報を取得する ..................................... 116</td>
<td></td>
</tr>
<tr>
<td>プロファイルの現在の状態を表示する .......................................... 116</td>
<td></td>
</tr>
<tr>
<td>補助的な状態の値 ............................................................... 118</td>
<td></td>
</tr>
<tr>
<td>プロファイルをアクティブまたは非アクティブにする .......................... 118</td>
<td></td>
</tr>
<tr>
<td>無線スキャンを実行して、使用可能な無線ネットワークに接続する .......... 121</td>
<td></td>
</tr>
<tr>
<td>NWAM ネットワーク構成のトラブルシューティング ............................ 122</td>
<td></td>
</tr>
<tr>
<td>すべてのネットワーク接続の現在の状態を監視する ....................... 122</td>
<td></td>
</tr>
<tr>
<td>ネットワークインタフェース構成に関する問題のトラブルシューティング ..... 123</td>
<td></td>
</tr>
<tr>
<td>6 NWAM グラフィカルユーザーインタフェースについて ............................ 125</td>
<td></td>
</tr>
<tr>
<td>NWAM グラフィカルユーザーインタフェースの紹介 .............................. 125</td>
<td></td>
</tr>
<tr>
<td>デスクトップから NWAM GUI にアクセスする ................................... 126</td>
<td></td>
</tr>
<tr>
<td>NWAM CLI と NWAM GUI の相違点 ........................................... 127</td>
<td></td>
</tr>
<tr>
<td>NWAM GUI の機能コンポーネント ............................................ 128</td>
<td></td>
</tr>
<tr>
<td>デスクトップから NWAM と対話する .......................................... 131</td>
<td></td>
</tr>
<tr>
<td>ネットワーク接続のステータスの確認 ........................................ 131</td>
<td></td>
</tr>
<tr>
<td>デスクトップからネットワーク接続を制御する ................................ 133</td>
<td></td>
</tr>
<tr>
<td>お気に入りの無線ネットワークに加入して管理する .......................... 135</td>
<td></td>
</tr>
</tbody>
</table>
目次

Ⅶ 無線ネットワークに加入する方法 ................................. 136
お気に入りのネットワークを管理する ............................ 137
ネットワークプロファイルの管理 ........................................... 137
  「ネットワーク設定」ダイアログについて ......................... 138
ネットワークプロファイルに関する情報を表示する .................. 140
あるネットワークプロファイルから別のネットワークプロファイルに切り替え る ................................................................. 140
ネットワークプロファイルを追加または削除する .................... 141
ネットワークプロファイルの編集 ............................................ 141
優先グループを操作する .................................................. 142
場所を作成および管理する .............................................. 144
場所の編集 ................................................................................ 146
外部ネットワーク識別者について .......................................... 147
  「ネットワーク修飾子」ダイアログについて ......................... 148
Ⅶ コマンド行の ENM を追加する方法 ................................. 149

パートⅡ データリンクとインタフェース構成 ......................... 151

7 プロファイルのデータリンクおよびインタフェース構成コマンドの使用 .......... 153
プロファイルベースのネットワーク構成の特徴 .................................. 153
プロファイルと構成ツール .................................................... 154
  ▼ ネットワーク管理モードを判定する方法 ........................... 155
次の手順 .................................................................................. 156

8 データリンクの構成と管理 .................................................. 157
データリンクの構成(タスク) .................................................. 157
dladm コマンド ................................................................. 158
  ▼ データリンクの名前を変更する方法 .................................. 160
  ▼ データリンクの物理属性に関する情報を表示する方法 ............... 161
  ▼ データリンクの情報を表示する方法 .................................. 162
  ▼ データリンクを削除する方法 ........................................... 163
データリンクプロバティーの設定 ............................................ 164
  dladm コマンドを使用したデータリンクプロバティーの設定 .......... 165
データリンクに関するその他の構成タスク .................................. 173

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
9 IPインタフェースの構成 .................................................................................................................. 179
  IPインタフェースの構成について .................................................................................................. 179
  ipadmコマンド ....................................................................................................................... 179
  IPインタフェースの構成(タスク) ............................................................................................... 181
  ▼SPARC:インタフェースのMACアドレスが一意であることを確認する方法 ........................................ 181
  IPインタフェースの構成 ........................................................................................................... 183
  ▼IPインタフェースを構成する方法 ........................................................................................... 183
    IPアドレスのプロパティの設定 .............................................................................................. 188
    IPインタフェースのプロパティの設定 ...................................................................................... 189
    プロトコルのプロパティの管理 ............................................................................................... 193
    TCP/IPプロパティの設定 ....................................................................................................... 193
  IPインタフェースとアドレスの監視 ........................................................................................... 198
  ▼ネットワークインタフェースに関する情報を取得する方法 .......................................................... 198
  インタフェース構成のトラブルシューティング ........................................................................... 202
  ipadmコマンドが動作しない。 .................................................................................................. 202
  ipadm create-addr コマンドでIPアドレスを割り当てることができない。 .............................. 203
  IPアドレスの構成中に cannot create address object: Invalid argument provided というメッセージが表示される。 ................................................................. 203
  IPインタフェースの構成中に cannot create address: Persistent operation on temporary object というメッセージが表示される。 ................................................................. 204
  比較表: ipadmコマンドとその他のネットワークコマンド .......................................................... 204
  ifconfigコマンドのオプションとipadmコマンドのオプション .................................................. 204
  nddコマンドのオプションとipadmコマンドのオプション ................................................................ 206

10 Oracle Solaris 上での無線インタフェース通信の構成 ............................................................... 209
  WiFi通信のタスクマップ ......................................................................................................... 209
  WiFiインタフェース上の通信 .................................................................................................... 210
    WiFiネットワークの検索 ......................................................................................................... 210
    WiFi通信の計画 .................................................................................................................... 211
  Oracle Solarisシステム上でのWiFiの接続および使用 .................................................................... 212
  ▼WiFiネットワークに接続する方法 ........................................................................................ 212
  ▼WiFiリンクを監視する方法 ..................................................................................................... 217
目次

セキュリティー保護された WiFi 通信 ................................................................. 218
 ▼ 暗号化された WiFi ネットワーク接続を設定する方法 ........................................ 219

11 ブリッジを管理する ................................................................................. 223
ブリッジングの概要 ....................................................................................... 223
 リンクプロパティー ..................................................................................... 227
 STP デーモン ............................................................................................... 229
 TRILL デーモン ............................................................................................ 229
 ブリッジをデバッグする ............................................................................. 230
 その他のブリッジ動作 ............................................................................... 231
 ブリッジの構成例 ....................................................................................... 233
 ブリッジの管理 (タスクマップ) ................................................................. 234
 ▼ 構成されているブリッジに関する情報を表示する方法 .................................. 235
 ▼ ブリッジリンクに関する構成情報を表示する方法 ...................................... 237
 ▼ ブリッジを作成する方法 ....................................................................... 237
 ▼ ブリッジの保護タイプを変更する方法 .................................................... 239
 ▼ 既存のブリッジに 1 つ以上のリンクを追加する方法 ...................................... 239
 ▼ ブリッジからリンクを削除する方法 ......................................................... 240
 ▼ システムからブリッジを削除する方法 ..................................................... 240

12 リンク集約の管理 .................................................................................. 241
リンク集約の概要 ......................................................................................... 241
 リンク集約の基本 ....................................................................................... 242
 バックツーバックリンク集約 ....................................................................... 243
 ポリシーと負荷分散 ................................................................................... 244
 集約モードとスイッチ ................................................................................ 245
 リンク集約の要件 ....................................................................................... 245
 リンク集約のための柔軟な名前 .................................................................. 246
 リンク集約の管理 (タスクマップ) ................................................................ 246
 ▼ リンク集約を作成する方法 ................................................................. 246
 ▼ 集約を変更する方法 ............................................................................... 249
 ▼ 集約にリンクを追加する方法 .................................................................. 249
 ▼ 集約からリンクを削除する方法 .............................................................. 250
 ▼ 集約を削除する方法 ............................................................................... 251
13 VLANの管理 ................................................................. 253
  仮想ローカルエリアネットワークの管理 .................................................. 253
  VLAN トポロジの概要 ............................................................................. 254
  VLAN の管理（タスクマップ） ................................................................. 256
  ネットワーク上の VLAN の計画 ............................................................. 257
  VLAN の構成 ......................................................................................... 258
  レガシーデバイス上の VLAN ................................................................. 262
  VLAN 上でのその他の管理タスクの実行 ............................................... 263
  カスタマイズされた名前を使用しているときのネットワーク構成タスクの結
  合 ............................................................................................................. 265

14 IPMP の紹介 ...................................................................................... 269
  IPMP の新機能 ....................................................................................... 269
  IPMP の配備 ........................................................................................... 270
    IPMP を使用しなければならない理由 .................................................. 270
    IPMP を使用する必要がある場合 ........................................................ 271
    IPMP とリンク集約の比較 .................................................................. 272
    IPMP 構成での柔軟なリンク名の使用 .................................................. 273
    IPMP の動作方法 ................................................................................ 274
  Oracle Solaris の IPMP コンポーネント .................................................. 280
  IPMP インタフェース構成のタイプ ......................................................... 281
  IPMP アドレス指定 ................................................................................ 282
    IPv4 検査用アドレス .......................................................................... 283
    IPv6 検査用 IP アドレス .................................................................... 283
  IPMP での障害および修復の検出 ............................................................ 284
    IPMP の障害検出の種類 .................................................................... 284
    物理インタフェースの回復検出 ............................................................. 287
  IPMP と動的再構成 ................................................................................ 288
    新しい NIC の接続 .............................................................................. 289
    NIC の切断 .......................................................................................... 290
    NIC の交換 ........................................................................................ 290
  IPMP の用語と概念 ................................................................................ 291

15 IPMP の管理 ...................................................................................... 301
  IPMP 構成のタスクマップ .................................................................... 301
# 目次

<table>
<thead>
<tr>
<th>項目</th>
<th>頁碼</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPMP グループの作成および構成（タスクマップ）</td>
<td>301</td>
</tr>
<tr>
<td>IPMP グループの維持（タスクマップ）</td>
<td>302</td>
</tr>
<tr>
<td>プローブベースの障害検出の構成（タスクマップ）</td>
<td>303</td>
</tr>
<tr>
<td>IPMP グループの監視（タスクマップ）</td>
<td>303</td>
</tr>
<tr>
<td>IPMP グループの構成</td>
<td>304</td>
</tr>
<tr>
<td>▼ IPMP グループの計画を立てる方法</td>
<td>304</td>
</tr>
<tr>
<td>▼ DHCP を使用して IPMP グループを構成する方法</td>
<td>306</td>
</tr>
<tr>
<td>▼ アクティブ・アクティブ IPMP グループを手動で構成する方法</td>
<td>309</td>
</tr>
<tr>
<td>▼ アクティブ・スタンバイ IPMP グループを手動で構成する方法</td>
<td>310</td>
</tr>
<tr>
<td>IPMP グループの維持</td>
<td>312</td>
</tr>
<tr>
<td>▼ IPMP グループにインタフェースを追加する方法</td>
<td>312</td>
</tr>
<tr>
<td>▼ IPMP グループからインタフェースを削除する方法</td>
<td>313</td>
</tr>
<tr>
<td>▼ IP アドレスを追加または削除する方法</td>
<td>314</td>
</tr>
<tr>
<td>▼ インタフェースを 1 つの IPMP グループから別のグループに移動する方法</td>
<td>315</td>
</tr>
<tr>
<td>▼ IPMP グループを削除する方法</td>
<td>316</td>
</tr>
<tr>
<td>プローブベースの障害検出のための構成</td>
<td>316</td>
</tr>
<tr>
<td>▼ 検査信号ベースの障害検出のターゲットシステムを手動で指定する方法</td>
<td>317</td>
</tr>
<tr>
<td>▼ 使用する障害検出手法を選択する方法</td>
<td>318</td>
</tr>
<tr>
<td>▼ IPMP デーモンの動作を構成する方法</td>
<td>319</td>
</tr>
<tr>
<td>動的再構成を使用した IPMP 構成の回復</td>
<td>320</td>
</tr>
<tr>
<td>▼ 障害が発生した物理カードを交換する方法</td>
<td>320</td>
</tr>
<tr>
<td>IPMP 情報の監視</td>
<td>322</td>
</tr>
<tr>
<td>▼ IPMP のグループ情報を取得する方法</td>
<td>322</td>
</tr>
<tr>
<td>▼ IPMP のデーダアドレス情報を取得する方法</td>
<td>323</td>
</tr>
<tr>
<td>▼ グループベースとなる IP インタフェースに関する情報を取得する方法</td>
<td>324</td>
</tr>
<tr>
<td>▼ IPMP のプローブターゲット情報を取得する方法</td>
<td>326</td>
</tr>
<tr>
<td>▼ IPMP のプローブを監視する方法</td>
<td>328</td>
</tr>
<tr>
<td>▼ スクリプト内で ipmpstat コマンドの出力をカスタマイズする方法</td>
<td>329</td>
</tr>
<tr>
<td>▼ ipmpstat コマンドのマシンによる解析が可能な出力を生成する方法</td>
<td>330</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16</th>
<th>LLDP によるネットワーク接続情報の交換</th>
<th>331</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Solaris での LLDP の概要</td>
<td>331</td>
<td></td>
</tr>
<tr>
<td>LLDP 実装のコンポーネント</td>
<td>331</td>
<td></td>
</tr>
<tr>
<td>LLDP エージェントの機能</td>
<td>333</td>
<td></td>
</tr>
<tr>
<td>LLDP エージェントの動作方法の構成</td>
<td>333</td>
<td></td>
</tr>
</tbody>
</table>

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
目次

パートⅢ ネットワーク仮想化およびリソース管理 .................................................. 345

17 ネットワーク仮想化およびリソース制御の紹介(概要) ........................................... 347
ネットワーク仮想化と仮想ネットワーク .......................................................... 347
内部仮想ネットワークの構成要素 ................................................................. 348
仮想ネットワークを実装すべきユーザー .......................................................... 350
リソース制御について .................................................................................. 351
帯域幅の管理とフロー制御のしくみ ................................................................. 351
ネットワーク上のリソースの割り当て制御と帯域幅の管理 ................................ 352
リソース制御機能を実装すべきユーザー .......................................................... 354
ネットワーク仮想化およびリソース制御のための可観測性機能 ................................ 354

18 ネットワーク仮想化およびリソース制御の計画 .................................................. 357
ネットワーク仮想化およびリソース制御のタスクマップ .................................... 357
仮想ネットワークの計画および設計 ................................................................. 358
1つのシステム上の基本的な仮想ネットワーク ............................................... 358
1つのシステム上のプライベート仮想ネットワーク ......................................... 360
参照先 ........................................................................................................... 362
ネットワークリソースに対する制御の実装 ...................................................... 362
従来のネットワークに対するインタフェースベースのリソース制御 ..................... 364
仮想ネットワークに対するフロー制御 ............................................................. 365
仮想ネットワーク上のアプリケーションに対する使用ポリシーを作成する方法 .................................................. 366
仮想ネットワークのためのサービスレベル契約を作成する方法 .......................... 366
| 19 | 仮想ネットワークの構成(タスク) ................................................................. | 369 |
|    | 仮想ネットワークのタスクマップ .................................................................. | 369 |
|    | Oracle Solaris でのネットワーク仮想化のコンポーネントの構成 .................. | 370 |
|    | ▼ 仮想ネットワークインタフェースを作成する方法 ........................................ | 371 |
|    | ▼ etherstub を作成する方法 ................................................................... | 373 |
|    | VNIC とゾーンの操作 .............................................................................. | 375 |
|    |   VNIC とともに使用するための新しいゾーンの作成 ................................. | 375 |
|    |   既存のゾーンの構成を VNIC を使用するように変更する ......................... | 380 |
|    |   プライベート仮想ネットワークの作成 .................................................. | 385 |
|    | ▼ ゾーンを削除することなく仮想ネットワークを削除する方法 ....................... | 387 |
| 20 | 仮想化環境でのリンク保護の使用 .............................................................. | 389 |
|    |  リンク保護の概要 ............................................................................... | 389 |
|    |   リンク保護タイプ ............................................................................. | 390 |
|    |  リンク保護の構成(タスクマップ) .......................................................... | 391 |
|    | ▼ リンク保護メカニズムを有効にする方法 ............................................... | 392 |
|    | ▼ リンク保護を無効にする方法 ............................................................ | 392 |
|    | ▼ IP のなりすましからの保護のための IP アドレスを指定する方法 ............... | 392 |
|    | ▼ リンク保護構成を表示する方法 ........................................................ | 393 |
| 21 | ネットワーククリソースの管理 .................................................................. | 395 |
|    |  ネットワーククリソースの管理の概要 ................................................... | 395 |
|    |  リソース制御のためのデータリンクプロパティ ....................................... | 395 |
|    |  フローを使用することによるネットワーククリソースの管理 .................... | 396 |
|    |  ネットワーククリソースの管理のためのコマンド .................................... | 397 |
|    | ネットワーククリソースの管理(タスクマップ) .......................................... | 398 |
|    | データリンク上のリソースの管理 ............................................................... | 399 |
|    |   送信リングと受信リング .................................................................. | 399 |
|    |   ブールと CPU ................................................................................... | 414 |
|    |  フロー上のリソースの管理 ................................................................. | 419 |
|    |  ネットワーク上のフローの構成 ............................................................ | 420 |
| 22 | ネットワークトラフィックとリソース使用状況の監視 ................................... | 425 |
|    | ネットワークトラフィックフローの概要 ................................................... | 425 |
目次

トラフィックとリソース使用の監視(タスクマップ) ............................................................ 428
リンク上のネットワークトラフィックに関する統計情報の収集 ........................................ 429
  ▼ ネットワークトラフィックに関する基本的な統計情報を取得する方法 .................... 430
  ▼ リンク使用状況に関する統計情報を取得する方法 ............................................... 432
  ▼ 設計上のネットワークトラフィックに関する統計情報を取得する方法 .................... 433
フロー上のネットワークトラフィックに関する統計情報の収集 ........................................... 435
  ▼ フローに関する統計情報を取得する方法 .............................................................. 436
ネットワークアカウンティングの設定 .......................................................................... 437
  ▼ 拡張ネットワークアカウンティングを構成する方法 .................................................. 438
  ▼ ネットワークトラフィックに関する履歴統計情報を取得する方法 ............................ 439

用語集 .............................................................................................................................. 443

索引 ....................................................................................................................................... 453
はじめに

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化へようこそ。このドキュメントは、Oracle Solaris システム管理に関する重要な情報を扱った全14巻セットの一部です。このドキュメントの記述は、Oracle Solarisがインストール済みであることが前提です。さらに、ネットワークを構成できる状態であり、そのネットワークに必要なネットワークソフトウェアを構成できる状態である必要があります。

注 - この Oracle Solaris のリリースでは、SPARC および x86 系列のプロセッサアーキテクチャーを使用するシステムをサポートしています。サポートされるシステムは、Oracle Solaris OS: Hardware Compatibility Lists に記載されています。本書では、プラットフォームにより実装が異なる場合は、それを特記します。

本書の x86 関連する用語については、次を参照してください。
- x86 は、64 ビットおよび32ビットのx86互換製品系列を指します。
- x64 は特に64ビットx86互換CPUを指します。
- 「32ビットx86」は、x86をベースとするシステムに関する32ビット特有の情報を指します。

サポートされるシステムについては、Oracle Solaris OS: Hardware Compatibility Listsを参照してください。

対象読者

このドキュメントは、Oracle Solaris が動作しており、ネットワークに構成されているシステムの管理を行うユーザーを対象としています。このマニュアルを利用するにあたっては、UNIXのシステム管理について少なくとも2年の経験が必要です。UNIXシステム管理のトレーニングコースに参加することも役に立ちます。
## Solaris システム管理マニュアルセットの構成
システム管理ガイドセットに含まれる各ガイドとその内容は、次のとおりです。

<table>
<thead>
<tr>
<th>マニュアルのタイトル</th>
<th>トピック</th>
</tr>
</thead>
<tbody>
<tr>
<td>『SPARC プラットフォームでの Oracle Solaris のブートおよびシャットダウン』</td>
<td>システムのブートおよびシャットダウン、ブートサービスの管理、ブート動作の変更、ZFSからのブート、ブートアーカイブの管理、およびSPARC プラットフォーム上でのブートのトラブルシューティング</td>
</tr>
<tr>
<td>『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』</td>
<td>システムのブートおよびシャットダウン、ブートサービスの管理、ブート動作の変更、ZFSからのブート、ブートアーカイブの管理、およびx86 プラットフォーム上でのブートのトラブルシューティング</td>
</tr>
<tr>
<td>『Oracle Solaris の管理: 一般的なタスク』</td>
<td>Oracle Solaris コマンドの使用、システムのブートとシャットダウン、ユーザーアカウントとグループの管理、サービスの管理、ハードウェア障害、システム情報、システムリソース、およびシステム性能、ソフトウェアの管理、印刷、コンソールと端末、およびシステムとソフトウェアの問題の障害追跡</td>
</tr>
<tr>
<td>『Oracle Solaris の管理: デバイスとファイルシステム』</td>
<td>リムーバブルメディア、ディスクとデバイス、ファイルシステム、およびデータのバックアップと復元</td>
</tr>
<tr>
<td>『Oracle Solaris の管理: IP サービス』</td>
<td>TCP/IP ネットワーク管理、IPv4 およびIPv6 アドレスの管理、DHCP、IPsec、IKE、IP フィルタ、およびIPQoS</td>
</tr>
<tr>
<td>『Oracle Solaris Administration: Naming and Directory Services』</td>
<td>DNS、NIS、およびLDAP ネームサービスおよびディレクトリサービス(NISからLDAPへの移行を含む)</td>
</tr>
<tr>
<td>『Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化』</td>
<td>WiFi ワイヤレスを含む自動および手動のIP インターフェース構成、ブリッジ、VLAN、集積体、LLDP、およびIPMP の管理、仮想NICとリソース管理</td>
</tr>
<tr>
<td>『Oracle Solaris のシステム管理 (ネットワークサービス)』</td>
<td>Web キャッシュサーバー、時間関連サービス、ネットワークファイアルシステム(NFS とautofs)、メール、SLP、およびPPP</td>
</tr>
<tr>
<td>『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』</td>
<td>リソース管理機能(利用可能なシステムリソースをアプリケーションが使用する方法を制御できる)、Oracle Solaris ゾーンソフトウェア区分技術(オペレーティングシステムのサービスを仮想化してアプリケーション実行用の隔離された環境を作成する)、およびOracle Solaris 10 ゾーン(Oracle Solaris 11 カーネル上で実行される Oracle Solaris 10 環境をホストする)</td>
</tr>
</tbody>
</table>
マニュアルのタイトル | トピック
---|---
『Oracle Solarisの管理:セキュリティサービス』 | 監査、デバイス管理、ファイルセキュリティ、BART、Kerberosサービス、PAM、暗号化フレームワーク、鍵管理、特権、RBAC、SASL、Secure Shell、およびウィルススキャン
『Oracle Solaris Administration: SMB and Windows Interoperability』 | SMB サービス (SMB クライアントが SMB 共有を利用できるように Oracle Solaris システムを構成できるようにする)、SMB クライアント (SMB 共有へのアクセスを可能にする)、およびネイティブの識別情報マッピングサービス (Oracle Solaris システムと Windows システム間でユーザーとグループの識別情報をマップできるようにする)
『Oracle Solaris の管理:ZFS ファイルシステム』 | ZFS ストレージプールおよびファイルシステムの作成と管理、スナップショット、クローン、バックアップ、アクセス制御リスト (ACL) による ZFS ファイルの保護、ゾーンがインストールされた Solaris システム上での ZFS の使用、エミュレートされたボリューム、およびトラブルシューティングとデータ回復
『Trusted Extensions構成と管理』 | Trusted Extensions に固有のシステムのインストール、構成、および管理
『Oracle Solaris 11 セキュリティガイドライン』 | Oracle Solaris システムのセキュリティー保護、およびゾーン、ZFS、Trusted Extensionsなどのセキュリティ機能の使用シナリオ
『Oracle Solaris 10 から Oracle Solaris 11 への移行』 | システム管理情報および Oracle Solaris 10 から Oracle Solaris 11 への移行例を提供します。移行例には、インストール、デバイス、ディスク、ファイルシステムの管理、ソフトウェアの管理、ネットワーク、システムの管理、セキュリティー、仮想化、デスクトップ機能、ユーザーアカウントの管理、ユーザー環境をエミュレートしたボリューム、およびトラブルシューティングとデータ回復が含まれます。

関連するサードパーティーサイト情報

このドキュメントにはオラクル社およびその関連会社が所有または管理しないWebサイトへのリンクが含まれている場合があります。
注 - オラクル社およびその関連会社は、それらのWebサイトのアクセシビリティに関しての評価や言及は行っておりません。このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

Oracle サポートへのアクセス


表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

<table>
<thead>
<tr>
<th>表 P-1 表記上の規則</th>
<th>例</th>
</tr>
</thead>
<tbody>
<tr>
<td>AaBbCc123</td>
<td>コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>変数を示します。実際に使用する特定の名前または値を置き換えるます。</td>
</tr>
<tr>
<td>『』</td>
<td>参照する書名を示します。</td>
</tr>
<tr>
<td>「 」</td>
<td>参照する章、節、ボタンやメニュー名、強調する単語を示します。</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
Oracle Solaris OSに含まれるシェルで使用する、UNIXのデフォルトのシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例に示されるデフォルトのシステムプロンプトは、Oracle Solarisのリリースによって異なります。

- **C シェル**
  ```
  machine_name% command y\n [filename]
  ```

- **C シェルのスーパーユーザー**
  ```
  machine_name# command y\n [filename]
  ```

- **Bash シェル、Korn シェル、および Bourne シェル**
  ```
  $ command y\n [filename]
  ```
  ```
  # command y\n [filename]
  ```

[] は省略可能な項目を示します。上記の例では、filename は省略してもよいことを示しています。

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

[] は区切り文字（セパレータ）です。この文字で分割されている引数のうち1つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します（例：Shiftキーを押します）。ただし、キーボードによってはEnterキーがReturnキーの動作をします。

ダッシュ（-）は2つのキーを同時に押すことを示します。たとえば、Ctrl-DはControlキーを押したままDキーを押すことを意味します。
この章では、Oracle Solaris でのネットワーク管理について紹介します。インタフェース、インタフェースが構成されているデータリンク、およびネットワークデバイスのベースとなる相互関係について説明します。また、データリンクの柔軟な名前でのサポートについても詳しく説明します。

この Oracle Solaris リリースでのネットワーク構成

このリリースでのネットワーク構成の方法が、以前の Oracle Solaris リリースと次の点で異なることに注意してください。

ネットワーク構成はプロファイルで管理されます。システムで機能する構成のタイプは、どのネットワーク構成プロファイルがアクティブなのかによって異なります。パート I 「Network Auto-Magic」 を参照してください。

ネットワークスプットの第２層のデータリンクは、dladm コマンドを使用して管理されます。このコマンドは、データリンクプロパティを構成するための以前の ifconfig コマンドオプションに代わるもので、したがって、リンク接続、VLAN、および IP トンネルの構成も変更されています。第８章「データリンクの構成と管理」、第１２章「リンク接続の管理」、および第１３章「VLAN の管理」を参照してください。『Oracle Solaris の管理：IP サービス』の第６章「IP トンネルの構成」も参照してください。

データリンク名はハードウェアデバイスにバインドされなくなりました。したがって、データリンクにはデフォルトで net0 や net1などの汎用リンク名が割り当てられています。26ページの「ネットワークデバイスとデータリンク名」を参照してください。

ネットワークスプットの第３層のIPインタフェースは、ipadm コマンドを使用して管理されます。このコマンドは、IP インタフェースを構成するための以前の ifconfig コマンドオプションに代わるもので、第 9 章「IP インタフェースの構成」を参照してください。
Oracle Solarisのネットワークスケルトン

IPMPグループはIPインタフェースとして実装されているため、同様にipadmコマンドを用いて構成されます。さらに、IPMP関連の情報や統計情報を取得できるipmpstatを導入されています。第14章「IPMPの紹介」および第15章「IPMPの管理」を参照してください。

仮想化はネットワークデバイスレベルで実装されています。したがって、効率性を高めるために、VNICを構成して、ネットワークリソースの使用を管理できます。パートⅢ「ネットワーク仮想化およびリソース管理」を参照してください。

Oracle Solarisのネットワークスケルトン

ネットワークインタフェースによって、システムとネットワークが接続されます。これらのインタフェースは、システムのハードウェアデバイスインスタンスに順に対応するデータリンク上に構成されます。ネットワークハードウェアデバイスは、ネットワークインタフェースカード（NIC）またはネットワークアダプタとも呼ばれます。NICは、システム購入時から組み込まれていて、すでに備わっている場合もあります。ただし、別のNICを購入して、システムに追加することもできます。一部のNICには、カードに備えられている単一のインタフェースしか持たないものがあります。その他のブランドは、ネットワーク操作を行うために構成できる複数のインタフェースを持っていることがあります。

現在のネットワークスケルトンモデルでは、ソフトウェア層のインタフェースおよびリンクはハードウェア層のデバイスに構築されます。具体的には、ハードウェア層のハードウェアデバイスインスタンスには、データリンク層の対応するリンク、およびインタフェース層の構成済みインタフェースが存在します。次の図では、このようなネットワークデバイス、データリンク、およびIPインタフェース間の1対1の関係について説明します。

注 - TCP/IPスケルトンの詳細は、「System Administration Guide: IP Services」の第1章「Oracle Solaris TCP/IP Protocol Suite (Overview)」を参照してください。
この図は、ハードウェア層の2基のNIC（単一のデバイスインスタンスe1000g0にはe1000、複数のデバイスインスタンスqfe0・qfe3にはqfeを示しています。デバイスqfe0・qfe2は使用されません。デバイスe1000gとqfe3が使用され、対応するリンクe1000gとqfe3がデータリンク層に存在します。この図では、IPインタフェースにそれぞれベースとなるハードウェアe1000gおよびqfe3に基づいて名前が付けられています。これらのインタフェースをIPv4またはIPv6アドレスで構成すると、両方のタイプのネットワークトラフィックに対応できます。インタフェース層にループバックインタフェースlo0が存在することにも注意してください。このインタフェースは、たとえば、IPスタックが正常に機能していることをテストするときなどに使用されます。

スタックの各層で、さまざまな管理コマンドが使用されます。たとえば、dladm show-devコマンドは、システムにインストールされているハードウェアデバイスを
Oracle Solarisのネットワークスタック

一覧表示します。dladm show-link コマンドは、データリンク層のリンクに関する情報
を表示します。ifconfig コマンドは、インタフェース層のIPインタフェース構成
を表示します。

このモデルでは、デバイス、データリンク、およびインタフェースをバインドする1
対1の関係が存在します。この関係は、ネットワーク構成がハードウェア構成およ
びネットワークトポロジに依存することを意味します。ハードウェア層に、NICの
交換やネットワークトポロジの変更などの変更が実装されている場合は、インタ
フェースを再構成する必要があります。

Oracle Solaris 11 に導入されているネットワークスタックの実装では、ハード
ウェア、データリンク、およびインタフェースの各層間の基本関係が存続していま
す。ただし、ソフトウェア層はハードウェア層から分離されています。この分離に
より、ソフトウェアレベルでのネットワーク構成は、ハードウェア層のチップ
セットやネットワークトポロジにバインドされなくなりました。この実装では、次
のようにネットワーク管理の柔軟性が高くなっています。

- ネットワーク構成は、ハードウェア層で発生する可能性のある変更から分離され
  ます。ベースとなるハードウェアが削除されても、リンクおよびインタフェース
  の構成は保持されます。したがって、2基のNICタイプが同じであれば、同じ構
  成を交換用のNICに再適用できます。
- ネットワーク構成がネットワークハードウェア構成から分離されることによ
  り、カスタマイズリンク名をデータリンク層で使用することもできます。
- データリンク層の抽象化によって、複数ネットワークの抽象化や構成
  (VLAN、VNIC、物理デバイス、リンク集約、IPトンネルなど)が共通の管理エン
  ディティー（つまり、データリンク）に統合されます。

次の図は、このようなネットワーク構成をネットワークスタックに作成する方法を
示しています。
図1-2 ネットワークデバイス、リンク、およびインタフェースを示すネットワークスタック - Oracle Solaris 11 モデル

このイラストの構成については、31ページの「その他のリンクタイプの管理」で詳細に説明します。
ネットワークデバイスとデータリンク名

ネットワークデバイスとデータリンク名

管理者は管理の観点から、データリンク上にIPインタフェースを作成します。データリンクは、開放型システム間相互接続(OSI)モデルの第2層のリンクオブジェクトを表します。物理リンクはデバイスに直接関連付けられ、デバイス名を持ちます。原則的に、デバイス名はデバイスインスタンス名であり、ドライバ名とデバイスインスタンス番号で構成されます。インスタンス番号には、システムでそのドライバを使用するNICの数に応じて、0～nの値を付けることができます。

たとえば、ホストシステムとサーバーシステムの両方でプライマリNICとして使用されることが多いギガビットEthernetカードについて考えてみます。このNICの典型的なドライバ名はbgeやe1000gなどです。ギガビットEthernetインタフェースをプライマリNICとして使用する場合、デバイス名はbge0やe1000g0などです。他のドライバ名はnxeやnxgeなどです。

このOracle Solarisでは、デバイスインスタンス名はベースとなるハードウェアに引き続き依存します。ただし、これらのデバイス上のデータリンクは同様にバインドされていないため、意味のある名前を付けることができます。たとえば、管理者者はデバイスインスタンスe1000g0上のデータリンクにitops0という名前を割り当てることができます。このOracle Solarisでは、データリンクにはデフォルトで汎用名が付けられています。汎用名が付けられたデータリンクは対応するデバイスインスタンスとのマッピングを表示するには、dladm sho-physサブコマンドを使用します。

デフォルトの汎用リンク名

このOracle Solarisでは、インストール・パッチ・アップデートにおけるネットワークデバイスの物理ネットワークデバイスに汎用リンク名が割り当てられます。この名前の割り当てでは、net #という命名規則が使用されます(#はインスタンス番号)。このインスタンス番号は、net0、net1、net2のよう

汎用名または柔軟なリンク名は、次の例で示すようなネットワーク構成における利点があります。

- 単一のシステムでは、動的に再構成が簡単になります。特定のNIC向けに設定されたネットワーク構成は、別のNICに統合するときに継承できます。
- ゾーンの移行によって、ネットワーク設定に関する複雑さが少なくなりました。移行元システムのリンクが、移行前にゾーンに割り当てられていたリンクと同じ名前を共有している場合、移行されたシステムのゾーンではネットワーク構成が保持されます。したがって、移行後の追加のネットワーク構成は必要ありません。
システム構成 (SC) マニフェストでネットワーク構成が指定されている場合は、汎用の名付けスキームが役立ちます。一般に、すべてのシステムのプライマリネットワークデーテリンクには、net0 という名前が付けられています。したがって、net0 の構成を指定する複数のシステムで、汎用 SC マニフェストを使用できます。

データリンクの管理も柔軟になります。たとえば、図1–2 で示すように、データリンクで提供される特定の機能が反映されるように、データリンク名をさらにカスタマイズできます。

次の表は、ハードウェア (NIC)、デバイスインスタンス、リンク名、およびリンク上のインタフェース間の新しい対応関係を示しています。データリンク名は自動的に OS で割り当てられます。

<table>
<thead>
<tr>
<th>ハードウェア (NIC)</th>
<th>デバイスインスタンス</th>
<th>リンクに割り当てられた名前</th>
<th>IP インタフェース</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1000g</td>
<td>e1000g0</td>
<td>net0</td>
<td>net0</td>
</tr>
<tr>
<td>qfe</td>
<td>qfe1</td>
<td>net1</td>
<td>net1</td>
</tr>
</tbody>
</table>

表で示すように、デバイスインスタンス名はハードウェアに基づくままである一方で、データリンク名はインストール後に OS によって変更されています。

データリンクへの汎用名の割り当て

Oracle Solaris では、特定の条件に基づいて、自動的にすべてのデータリンクに汎用名が割り当てられます。すべてのデバイスは、net という同一の接頭辞を共有します。ただし、インスタンス番号は次の場合に基づいて割り当てられます。

物理ネットワークデバイスは、メディアタイプに従って順序が付けられ、特定のタイプが他よりも優先されます。メディアタイプは、次のような降順の優先度で順序が付けられます。
1. Ethernet
2. IP over IB (Infiniband デバイス)
3. Ethernet over IB
4. WiFi

デバイスがメディアタイプに従ってグループ化およびソートされたあとは、これらのデバイスは物理的な場所に基づいて、さらに順序が付けられ、ボード上のデバイスが周辺デバイスよりも優先されます。

メディアタイプと場所に基づいた優先度が高いデバイスには、小さいインスタンス番号が割り当てられます。
ネットワークデバイスとデータリンク名

条件に基づいて、下位のマザーボードやIO ボード、ホストブリッジ、PCIeルートコ
ンプレックス、バス、デバイス、および機能上の Ethernet デバイスが他のデバイス
よりもランクが上になります。

リンク名、デバイス、および場所の対応関係を表示するには、次のように dladm
show-phys コマンドを使用します。

```
# dladm show-phys -L
```

<table>
<thead>
<tr>
<th>LINK</th>
<th>DEVICE</th>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>net0</td>
<td>e1000g0</td>
<td>MB</td>
</tr>
<tr>
<td>net1</td>
<td>e1000g1</td>
<td>MB</td>
</tr>
<tr>
<td>net2</td>
<td>e1000g2</td>
<td>MB</td>
</tr>
<tr>
<td>net3</td>
<td>e1000g3</td>
<td>MB</td>
</tr>
<tr>
<td>net4</td>
<td>ibp0</td>
<td>MB/RISER0/PCIE0/PORT1</td>
</tr>
<tr>
<td>net5</td>
<td>ibp1</td>
<td>MB/RISER0/PCIE0/PORT2</td>
</tr>
<tr>
<td>net6</td>
<td>eoib2</td>
<td>MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2</td>
</tr>
<tr>
<td>net7</td>
<td>eoib4</td>
<td>MB/RISER0/PCIE0/PORT2/cloud-nm2gw-2/1A-ETH-2</td>
</tr>
</tbody>
</table>

汎用リンク名の割り当て方法をカスタマイズする

Oracle Solaris では、リンク名の割り当て時に net という接頭辞が使用されます。ただ
し、代わりに eth などのカスタム接頭辞も使用できます。必要であれば、中立的な
リンク名の自動割り当てを無効にすることもできます。

注意 - Oracle Solaris をインストールする前に、汎用リンク名を自動的に割り当てる方
法をカスタマイズする必要があります。インストール後は、既存の構成を破棄しな
いかぎりデフォルトのリンク名をカスタマイズすることはできません。

リンク名の自動割り当てを無効にしたり、リンク名の接頭辞をカスタマイズしたり
するには、自動インストール (AI) プログラムで使用される「システム構成」マニ
フェストで、次のプロパティを設定します。

```
<service name="network/datalink-management"
  version="1" type="service">
  <instance name="default enabled="true">
    <property_group name='linkname-policy' type='application'>
      <propval name='phys-prefix' type='astring'
        value='net'/>
    </property_group>
  </instance>
</service>
```

強調表示されているように、デフォルトでは phys-prefixがnetに設定されています。

名前の自動割り当てを無効にするには、phys-prefix に設定されている値をすべ
て削除します。名前の自動割り当てを無効にすると、関連するハードウェアドライ
バに基づいて、bge0やe1000g0などのデータリンク名が付けられます。
ネットワークデバイスとデータリンク名

- net以外の接頭辞を使用するには、phys-prefixの値としてethなどの新しい接頭辞を指定します。

phys-prefixに指定された値が無効な場合は、その値は無視されます。データリンクには、関連するハードウェアドライバに従ってbge0やe1000g0などの名前が付けられます。有効なリンク名の規則については、30ページの「有効なリンク名の規則」を参照してください。

アップグレードされたシステムでのリンク名

このOracle Solarisリリースが新規インストールされたシステムでは、自動的にnet0-netN-1の名前が付けられます。ここで、Nはネットワークデバイスの合計数を表します。

Oracle Solaris 11 Expressからアップグレードする場合、これも適用されません。このようにアップグレードされたシステムでは、アップグレード前のデータリンク名が保持されます。これらの名前は、デフォルトのハードウェアに基づく名前、またはアップグレード前に管理者がデータリンクに割り当てたカスタマイズされた名前です。さらに、これらのアップグレードされたシステムでは、あとで追加される新規ネットワークデバイスでも、中間的な名前を受け入れるのではなく、デフォルトのハードウェアに基づく名前が保持されます。アップグレードされたシステムでのこの動作によって、OSで割り当てられた中立的な名前が、その他のハードウェアに基づく名前やアップグレード前に管理者によって割り当てられたカスタマイズされた名前と混在しなくなります。

このOracle Solarisリリースのシステムでは、ハードウェアに基づく名前とOSで割り当てられたリンク名は、両方とも他の任意の名前で置き換えることができます。一般に、システムのネットワーク構成を作成するには、OSで割り当てられたデフォルトのリンク名で十分です。ただし、リンク名を変更することを選択した場合は、次の節で説明する重要な考慮点に注意してください。

ハードウェアに基づくリンク名を置き換える

システムのリンクにハードウェアに基づく名前が付けられている場合は、これらのリンク名を少なくとも汎用名に変更します。ハードウェアに基づくリンク名を保持すると、あとでこれらの物理デバイスが削除または交換される状況で混乱が生じる可能性があります。

たとえば、デバイスbge0に関連付けられたbge0というリンク名を保持するとします。すべてのリンク構成は、リンク名を参照することで実行されます。あとで、NIC bgeをNIC e1000gに交換するとします。新しいNIC e1000g0に以前のデバイスのリンク構成を適用するには、リンク名をbge0からe1000g0に再割り当てる必要があります。ハードウェアに基づくリンク名bge0に関連する別のNIC e1000g0と組み合わせると、混乱が生じる可能性があります。ハードウェアに基づかない名前を使用すれば、そのリンクを関連するデバイスと適切に区別することができます。
ネットワークデバイスとデータリンク名

リンク名を変更する際の注意点
ハードウェアに基づくリンク名を置き換えることは推奨されていますが、リンク名を変更する前に慎重に計画してください。デバイスのリンク名を変更しても、新しい名前が既存の関連するすべての構成に自動的に反映されるわけではありません。次の例は、リンク名を変更する際のリスクを示しています。

- 特定のリンクには、IP フィルタ構成の規則の一部が適用されます。リンク名を変更しても、振り分け規則は引き続きリンクの元の名前を参照します。したがって、リンク名を変更したあとは、これらの規則が予期したとおりに動作しなくなります。新しいリンク名を使用して、リンクに適用されるように振り分け規則を調整する必要があります。
- ネットワーク構成の情報をエクスポートする可能性について考慮してください。前述のとおり、OS で割り当てられたデフォルトの net # 名を使用すると、簡単に別のシステムにゾーンを移行して、ネットワーク構成をエクスポートできます。ターゲットシステムのネットワークデバイスに net0 や net1 などの汎用名を付けると、ゾーンは単純に、ゾーンに割り当てられたデータリンクと名前が一致するデータリンクのネットワーク構成を継承します。

したがって、一般的な規則として、データリンク名はランダムに変更しないでください。データリンク名を変更する際は、リンク名が変更されたあとに、すべてのリンク関連の構成が引き続き適用されていることを確認します。リンク名を変更すると影響を受ける可能性のある構成の一部は、次のとおりです。
- IP 振り分け規則
- 構成ファイルに指定された IP 構成 (/etc/dhcp.* など)
- Oracle Solaris 11 ゾーン
- autopush 構成

注 - リンク名を変更する際には、autopush 構成を変更する必要はありません。ただし、リンク名を変更したあとに、リンクごとの autopush プロパティによる構成の動作に注意する必要があります。詳細は、176 ページの「データリンクに STREAMS モジュールを設定する方法」を参照してください。

有効なリンク名の規則
リンク名を割り当てる際には、次の規則を確認してください。
- リンク名は、文字列と物理接続点 (PPA) 番号で構成されています。
- 名前は次の制約に従う必要があります。
  - 名前は 3-8 文字で構成されます。ただし、最大で 16 文字までです。
  - 名前に有効な文字は、英数字 (a-z, 0-9) と下線 (「_」) です。
その他のリンクタイプの管理

ネットワーク構成とネットワークハードウェア構成が分離されたことによって、その他のリンク構成タイプにも同様の柔軟性が導入されています。たとえば、仮想ローカルエリアネットワーク (VLAN)、リンク集約、および IP トンネルに管理用に選択された名前を割り当て、それらの名前を参照して構成することができます。ネットワーク構成が削除されていなければ、追加のネットワーク再構成も必要ないため、その他の関連タスク (動的再構成 (DR) を実行してハードウェアデバイスを交換するなど) の実行も簡単になります。

次の図は、デバイス、リンクタイプ、および対応するインタフェース間の相互関係を示しています。

この図では、システムで実行される特定の機能に従って、データリンクに video0 や sales2 などの名前が付けられています。図の目的は、柔軟なデータリンクに名前付けられたことを強調することです。ただし、OS で割り当てられたデフォルトの中立的な名前 (net0 など) の使用でも十分であり、推奨されています。

この図には、管理用に選択された名前をネットワーク設定で使用する方法のサンプルも示しています。

- net0 リンクには VLAN が構成されています。これらの VLAN には、sales1 や sales2 などのカスタマイズされた名前も割り当てられています。VLAN sales2 の IP インタフェースが plumb され、動作しています。
その他のリンクタイプの管理

- デバイスインスタンス qfe0 および qfe2 は、ビデオトラフィックの処理に使用されます。したがって、データリンク層の対応するリンクには、subvideo0 および subvideo1 という名前が割り当てられています。これらの 2 つのリンクは、ビデオフィードを処理するために集約されます。リンク集約にも同様に、video0 という独自のカスタマイズされた名前が存在します。
- 電子メールトラフィックを処理するために、別々のベースとなるハードウェア (el000g と qfe) に搭載された 2 つのインタフェース (net0 と net1) が IPMP グループ (itops0) としてまとめられています。

注-IPMP インタフェースはデータリンク層のリンクではありませんが、これらのインタフェースにも、リンクと同様にカスタマイズされた名前を割り当てることができます。IPMP グループの詳細は、第 14 章「IPMP の紹介」を参照してください。

- 2 つのインタフェース (VPN 接続用に構成された vpn1 および VPN ループバック操作用に構成された lo0) には、ベースとなるデバイスがありません。

この図で示すリンクおよびインタフェースの構成はすべて、ベースとなるハードウェアの構成から独立しています。たとえば、qfe カードが交換されても、ビデオトラフィック用の video インタフェースの構成は保持され、あとは交換用の NIC に適用できます。

次の図は、ブリッジ構成を示しています。2 つのインタフェース (net0 および videoagg0) がブリッジ bridge0 として構成されています。一方のインタフェースで受信されたパケットは、他方のインタフェースに転送されます。ブリッジの構成後も、両方のインタフェースを使用して VLAN および IP インタフェースを構成できます。
パート1

Network Auto-Magic

Network Auto-Magic (NWAM) は、システムの基本的なネットワーク構成を自動化する Oracle Solaris の機能です。これらの章で扱うトピックでは、NWAM アーキテクチャーのコンポーネント、および Oracle Solaris システムで自動的にネットワーク構成を実行するためにこれらのコンポーネントがどのように連携するかについて説明します。

このドキュメントでは主に、NWAM コマンド行ユーティリティーを使用してネットワーク構成を管理する方法に焦点を当てています。また、NWAM グラフィカルユーザーインタフェース (GUI) を使用してネットワークのステータスを表示および監視する方法、およびデスクトップから NWAM と対話する方法についても説明します。NWAM GUI を使用してネットワーク構成を監視および管理するための詳細な手順は、オンラインヘルプで検索できます。
Network Auto-Magic (NWAM) 機能では、起動時の有線または無線ネットワークへの接続やネットワーク接続ステータスのデスクトップへの通知などの、基本的なEthernetおよびWiFi構成を自動的に処理することで、基本的なネットワーク構成が簡略化されています。また、NWAMは、システム全体のネットワークプロファイアルの作成や管理などの、より複雑なネットワークタスクの一部を簡略化するように設計されています。たとえば、ネームサービス、IPフィルタ、およびIPセキュリティ（IPsec）の構成などのOracle Solarisの機能が含まれます。

この章で扱う内容は、次のとおりです。
- 38ページの「NWAM構成とは」
- 40ページの「NWAMを使用する場合」
- 41ページの「NWAM構成の動作」
- 42ページの「その他のOracle Solarisネットワーク技術を使用したNWAMの動作」
- 44ページの「ネットワーク構成タスクを検索する場所」

この章は、基本的なネットワークの概念を理解していて、従来のネットワークツールやコマンドを使用してネットワーク構成を管理した経験がいくらかあるユーザーおよびシステム管理者を対象としています。すぐにNWAMを使用してネットワーク構成を管理できる場合は、第4章「NWAMプロファイルの構成(タスク)」に進んでください。

Oracle Solarisでのネットワークインタフェースの管理ついての基本情報は、パートII「デ Bitsリングとインタフェース構成」を参照してください。
NWAM 構成とは

NWAM 構成は、システムのネットワーク構成をできる限り自動的に実行するために、連携して動作する複数のコンポーネントで構成されています。機動性に必要な焦点を当てるため、NWAMでは、さまざまなネットワークイベントやユーザーの要求に応じて、システムの構成を動的に変更できます。NWAMには、ネットワーク状況の変化（取り付けられたネットワークインタフェースが取り外された場合や、新しい無線ネットワークが使用可能になった場合など）に対処する動的な機能が備わっています。

NWAMによるネットワーク構成は、さまざまなタイプのプロファイルに関連付けられたプロパティーやその値で構成されます。これらは、構成オブジェクトと呼ばれることもあります。

これらのプロファイルおよび構成オブジェクトには、次の項目が含まれます。

- ネットワーク構成プロファイル（NCP）
  NCPには、ネットワークリンクおよびインタフェースの構成が指定されます。このプロファイルは、NWAM構成を構成する主要なプロファイルタイプの1つです。2つ目の主要なプロファイルタイプは、場所プロファイルです。
  システムでは常に、Automatic NCPと呼ばれるNCPが定義されます。このNCPは、ユーザーからの入力がない場合にアクティブになります。Automatic NCPは、システムで作成および保守され、変更や削除はできません。
  必要に応じて、ユーザーの定義の追加NCPを作成することもできます。Automatic NCPおよびユーザー定義のNCPの詳細は、50ページの「Automatic NCPとユーザー定義のNCPの説明」を参照してください。

- ネットワーク構成ユーティット（NCU）
  NCUとは、NCPを構成するプロパティーやすべて含める個々の構成オブジェクトです。基本的には、NCPは定義するNCUを格納するコンテナです。各NCUは、システムの個々のリンクまたはインタフェースに関連付けられます。NCUの詳細は、49ページの「NCUの説明」を参照してください。

- 場所
  場所プロファイルは、NWAM構成を構成する2つの主要なプロファイルタイプの1つです。場所には、システム全体のネットワーク構成（ネットワークサービス、ドメイン、IPフィルタ、IPsecの構成など）が指定されます。この情報は、システム全体のネットワーク構成に適用されるプロパティーセットで構成されます。システム定義とユーザー定義の両方の場所があります。場所プロファイルの詳細は、50ページの「場所プロファイルの説明」を参照してください。

- 外部ネットワーク修飾子（ENM）
  ENM とは、NWAM外部のアプリケーション（VPNアプリケーションなど）を管理するために使用されるプロファイルです。これらのアプリケーションでは、ネットワーク構成の変更および作成が可能です。nwamdデーモンは、ENMの
一部として指定されている条件に応じて、ENM をアクティブ化または非アクティブ化します。ENM の詳細は、51 ページの「ENM の説明」を参照してください。

- 既知の無線ローカルエリアネットワーク (WLAN)

既知の WLAN とは、システムに認識されている無線ネットワークに関する情報を監視および格納するために NWAN で使用される構成オブジェクトです。NWAN では、このような無線ネットワークの全リストが保持され、使用可能な無線ネットワークへの接続を試みる順序を決定する際に、このリストが参照されます。既知の WLAN の詳細は、52 ページの「既知の WLAN について」を参照してください。

**NWAM の機能コンポーネント**

NWAM は、次の機能コンポーネントで構成されています。

- **NWAM プロファイルリポジトリ－プロファイルリポジトリ**は、NWAM 構成データが格納される場所です。プロファイルリポジトリへのアクセス、リポジトリデモニオン netcfgd で管理されます。

NWAM プロファイルリポジトリには、NWAM が有効であるときのネットワーク構成のスナップショットが含まれています。このデータは、ネットワークの手動構成に戻す必要がある場合に備えて保存されています。詳細は、53 ページの「NWAM 構成データ」を参照してください。

- プロファイル構成プログラム (ユーザーアイナフェース)NWAM アーキテクチャ的には、コマンド行インタフェース (CLI) とグラフィカルユーザーインタフェース (GUI) の両方が用意されています。これらのインタフェースでは、同一ようなタスク、たとえば、プロファイルの作成と変更、プロファイルの有効化、およびプロファイルに関する情報についてのシステムへの問い合わせなどを実行できます。

NWAM CLI は、netcfg と netadm の 2 つの管理用コマンドで構成されています。netcfg コマンドを使用すると、プロファイルを作成および変更できます。このコマンドは、対話型モード、コマンド行モード、およびコマンドファイルモードで実行されます。netadm コマンドを使用すると、プロファイルの有効化または無効化、プロファイル状態に関するレポートの一覧表示などの、特定の操作を実行できます。詳細は、netcfg(1M) および netadm(1M) のマニュアルページを参照してください。

NWAM CLI を使用してプロファイルを作成および管理する手順については、第 4 章「NWAM プロファイルの構成(タスク)」および第 5 章「NWAM プロファイルの管理(タスク)」を参照してください。

NWAM GUI を使用すると、ネットワークプロファイルを作成および管理することもできます。GUI には、ネットワーク接続のステータスをデスクトップからすばやく表示して監視できる追加の機能があります。また、GUI には、現在のネットワークステータスの変化についてユーザーに警告する通知機能もありま
NWAMを使用する場合

す。通知機能は、GUIでのみ使用可能です。NWAM GUIの使用の詳細は、第6章「NWAM グラフィカルユーザーインタフェースについて」またはオンラインヘルプを参照してください。nwammgr(1M)およびnwammgr-properties(1M)のマニュアルページも参照してください。

ポリシーエンジンデーモン - nwamd デーモンはNWAMのポリシーコンポーネントです。このデーモンは複数の役割で機能し、プロファイリポジトリに格納されたプロファイルに基づいてネットワーク構成を管理します。また、現在のネットワーク状況に対応するアクティブにするプロファイルを決定し、そのプロファイルをアクティブにします。このタスクを完了するために、デーモンは複数のソースからの情報を統合します。nwamdデーモンが遂行する複数の役割については、71ページの「NWAM デーモンの概要」節で詳細に説明します。

ルポジトリデーモン - netcfgd デーモンは、プロファイルおよびその他の構成オブジェクトの構成データがすべて格納される共通のプロファイリリポジトリを制御します。netcfg コマンド、NWAM GUI、およびnwam デーモンはすべて、プロファイリポジトリへのアクセス要求を送信することによって、netcfgd デーモンと対話します。ルポジトリデーモンのジョブは、ルポジトリデータへのアクセスを試みるさまざまなプロセスが正しく許可されているかどうかを確認することです。このデーモンは、許可されていないプロセスによるアクセスの試みを禁止（破棄）します。詳細は、72ページの「NWAM ルポジトリデーモン (netcfgd) の説明」を参照してください。

NWAMライブラリインタフェース - libnwam ライブラリは、プロファイルリポジトリと対話するための機能インタフェースを提供します。これにより、プロファイルに関する情報をNWAMで読み取って変更することができます。

サービス管理機構(SMF)ネットワーク：NWAMで使用される複数のネットワークサービスは、すでにOracle Solarisの一部になっています。ただし、これらの既存サービスは一部変更され、NWAMに固有の新規サービスが導入されています。詳細は、72ページの「SMFネットワークサービス」を参照してください。

NWAMを使用する場合

一般に、作業環境や接続方式(有線または無線)を頻繁に変更する場合は、NWAMの自動ネットワーク構成機能を利用します。NWAMを使用すると、さまざまな設定(オフィス、自宅、または外出先)でネットワークに接続できるように、ユーザー定義のプロファイルを設定できます。NWAMは、ネットワーク環境を頻繁に変更する必要があるノートパソコンモデルおよびシステムのユーザーに役立つツールです。さらに、NWAM GUIを使用すると、従来のネットワークツールやコマンドよりも簡単に、静的IPの構成やWiFiネットワークへの接続を設定できます。

NWAMは、ネットワーク環境の変更(Ethernet接続の損失や、ネットワークインタフェースカード(NIC)の取り外しなど)に適応するために構成できます。
NWAM構成の動作

NWAMのデフォルト動作では、有線または無線ネットワークの基本構成がユーザー操作なしで「自動的に」実行されます。NWAMと対話する必要があるのは、無線ネットワークのセキュリティーキーやパスワードを入力するなど、システムから詳細な情報を要求された場合に限ります。

NWAMの自動構成は、次のイベントおよびアクティビティーで呼び出されます。

- Ethernetケーブルを接続または切断する
- WLANカードを接続または切断する
- 有線インタフェースまたは無線インタフェース(あるいは両方)が使用可能なときにシステムをブートする
- 有線インタフェースまたは無線インタフェース(あるいは両方)が使用可能でない(サポートされている)ときに保存停止から再開する
- DHCPリースを取得または解放する

NWAMコンポーネントは、次の場合でほかのコンポーネントと対話します。

- 常に、1つのNCPと1つの場所プロファイアルをシステムでアクティブにする必要があります。
- システムブート中に、ポリシーエンジンデーモンnwamdによって次の操作が実行されます。
  1. 現在アクティブなNCPについてサービスプロパティーを調査します。
  2. 1つ以上のIPアドレスが構成されるまで続行します。
  3. 場所プロファイアルの状況を確認します。
  4. ポリシーエンジンで指定された場所プロファイアルをアクティブにします。
  5. ネットワーク(複数可)を適宜構成します。

ネットワーク構成の変更を引き起こす可能性のあるイベントが発生すると、NWAMデーモンnwamdはさまざまな役割を果たし、次の操作を実行します。

1. nwamdはイベントハンドラとして、各イベントの発生時にそのイベントを検出します。
2. nwamdはプロファイアルデーモンとして、アクティブなプロファイアルを調査します。
3. nwamdは変更に応じて、ネットワーク(複数可)を適宜再構成する場合があります。
NWAM のデフォルト動作

ユーザー定義のネットワークプロファイルが存在しない場合、nwamd は次の 3 つのシステム定義のプロファイルに基づいて、ネットワーク構成を管理します。

- Automatic NCP
- 場所「Automatic」
- 場所「NoNet」

Automatic NCP では、次の基本ポリシーが実装されます。

- DHCP を使用して、使用可能な (接続されている) すべての Ethernet インタフェースを構成します。
- Ethernet インタフェースが接続されていない場合や、IP アドレスを取得できない場合は、1 つの無線インタフェースをアクティブにすると、既知の WLAN リストから最適な WLAN が選択され自動的に接続されます。または、ユーザーが無線ネットワーク接続を選択するまで待機します。
- 1 つ以上の IPv4 アドレスが取得されるまで、場所「NoNet」はアクティブのままでです。この場所プロファイルには、IP アドレスの取得に関連するデータ (DHCP および IPv6 autoconf メッセージ) のみを渡すという厳格な IP 振り分け規則があります。場所「NoNet」のすべてのプロパティは、アクティブ化の条件を除いて変更可能です。
- 1 つ以上の IPv4 アドレスがシステムのインタフェースのいずれかに割り当てられると、場所「Automatic」がアクティブになります。この場所プロファイルには、IP フィルタや IPsec の規則はありません。場所プロファイルには、DHCP サーバーから取得された DNS 構成データが適用されます。場所「NoNet」と同様に、場所「Automatic」のすべてのプロパティは、アクティブ化の条件を除いて変更可能です。
- システムで IPv4 アドレスが割り当てられていない場合は、常に場所「NoNet」が適用されます。1 つ以上の IPv4 アドレスが割り当てられている場合は、現在のネットワーク状況に最適なアクティブ化の規則を持つ場所プロファイルが選択されます。より適した一致が存在しない場合は、場所「Automatic」に戻ります。詳細は、58 ページの「NWAM プロファイルをアクティブにする方法」を参照してください。

その他の Oracle Solaris ネットワーク技術を使用した NWAM の動作

NWAM は、次に示すその他の Oracle Solaris ネットワーク技術を使用して動作します。

- ネットワーク仮想化
NWAM は、次のようなさまざまな Oracle Solaris のネットワーク仮想化を使用して動作します。

- 仮想マシン: Oracle VM Server for SPARC（以前の論理ドメイン）および Oracle VM VirtualBox
  NWAM は Oracle Solaris のホストとゲストの両方でサポートされています。NWAM は指定された仮想マシンに属するインタフェースのみを管理し、ほかの仮想マシンには干渉しません。

- Oracle Solaris ゾーンとスタックインスタンス
  NWAM は、大域ゾーンまたは排他的なスタック（非大域）ゾーンで動作します。

注 - NWAM は、共有スタックゾーンでは動作しません。

- VNIC
  現在の NWAM 実装では VNIC は管理されませんが、手動で作成された VNIC はリブートしても存続し、排他的なスタックゾーンへの割り当てなどのために作成できます。

- ブリッジ技術
  ブリッジ技術は別々のネットワークセグメントを接続する方式であり、単一のセグメントだけが使用されているかのように、接続されたノード間の通信を行うことができます。現在の NWAM 実装では、ブリッジ技術を使用したネットワーク構成は積極的にサポートされていませんが、この技術をシステムで使用する前に、NWAM 構成管理を無効にする必要はありません。

- 動的再構成とネットワーク構成プロファイル
  動的再構成（DR）およびホットプラグ機能がサポートされているシステムでは、これらの機能がすぐに使用されるのは、システム上のアクティブな NCP が「DefaultFixed」の場合に限ります。

これらのシステムで有効な NCP が「自動」またはユーザーが作成したその他の NCP である場合は、DR 操作を実行する前に、まず次の手順のいずれかを実行する必要があります。

- ネットワークサービスを停止します。この操作によって、システム上のすべてのネットワークインタフェースが停止します。したがって、サービスを停止するときは、システムコンソールを使用する必要があります。デバイスを削除または交換したら、サービスを再起動します。

- netcfg コマンドを使用して、アクティブな NCP の構成から IP インタフェースを削除します。次に、その IP インタフェースのベースとなるハードウェアデバイスの物理的な削除または交換ができます。該当する場合は、DR の完了後に IP インタフェースを再構成します。

- 従来のネットワークコマンドとユーティリティ
ネットワーク構成タスクを検索する場所

常に、システムでは従来のネットワーク構成か、NWAM ネットワーク構成のいずれかが使用されます。「DefaultFixed」の NCP が有効な場合は、システムでは従来のネットワーク構成が使用されます。この NCP が有効な場合

 NWAM でネットワーク構成が管理されている場合でも、コマンド行のネットワークユーティリティ dladm および ipadm を使用すれば、現在のネットワーク構成のコンポーネントを表示できます。

注：コマンド行ツールを使用してネットワーク構成を変更すると、その変更が NWAM で適用されるポリシーと競合する可能性があるため、サポートされていません。

- IP ネットワークマルチパス (IPMP)
現在、NWAM では IPMP の使用はサポートされていません。IPMP が使用されるようにネットワークを構成する前に、「DefaultFixed」の NCP が有効になっていることを確認します。

ネットワーク構成タスクを検索する場所

次の表には、ネットワーク構成のトピック、および詳細情報の記載場所を一覧表示します。

<table>
<thead>
<tr>
<th>ネットワークタスク</th>
<th>参照先</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWAM に関する詳細な概要情報を検索します。</td>
<td>第3章「NWAM 構成と管理 (概要)」</td>
</tr>
<tr>
<td>NWAM CLI を使用して、プロファイルと構成オブジェクトの作成、変更、および削除を行います。</td>
<td>第4章「NWAM プロファイルの構成 (タスク)」</td>
</tr>
<tr>
<td>NWAM CLI を使用して、管理者プロファイルおよび構成オブジェクトに関する情報を表示します。</td>
<td>第5章「NWAM プロファイルの管理 (タスク)」</td>
</tr>
<tr>
<td>デスクトップから NWAM GUI を使用して、ネットワークステータスに関する情報の表示、ネットワーク接続の切り替え、およびプロファイルと構成オブジェクトの作成と変更を行います。</td>
<td>第6章「NWAM グラフィカルユーザーアンタフェースについて」およびオンラインヘルプ</td>
</tr>
</tbody>
</table>

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
ネットワーク構成タスクを検索する場所

<table>
<thead>
<tr>
<th>ネットワークタスク</th>
<th>参照先</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWAM ネットワーク構成モードと従来のネットワーク構成モードを切り替えます。</td>
<td>112ページの「ネットワーク構成の管理」</td>
</tr>
<tr>
<td>従来のネットワークツールおよびコマンドを使用して、ネットワーク構成を管理します。</td>
<td>第8章「データリンクの構成と管理」および第9章「IPインタフェースの構成」</td>
</tr>
<tr>
<td>仮想ネットワークを構成および管理します。</td>
<td>第17章「ネットワーク仮想化およびリソース制御の紹介(概要)」</td>
</tr>
</tbody>
</table>

第2章・NWAMの紹介
第3章

NWAM 構成と管理 (概要)

この章では、NWAM の構成および管理プロセスの背景と概要について説明します。NWAM でネットワーク構成を簡素化および自動化するために使用されるプロファイルの実装についても、詳細に説明します。

この章で扱う内容は、次のとおりです。

- 47 ページの「NWAM 構成の概要」
- 53 ページの「NWAM 構成データ」
- 58 ページの「NWAM プロファイルをアクティブにする方法」
- 63 ページの「netcfg を使用してプロファイルを構成する」
- 69 ページの「netadm コマンドを使用してプロファイルを管理する」
- 71 ページの「NWAM デーモンの概要」
- 72 ページの「SMF ネットワークサービス」
- 73 ページの「NWAM セキュリティーの概要」

NWAM 構成の概要

NWAM は、優先されるプロバティー値をプロファイル的形式でシステム上に格納することによって、ネットワーク構成を管理します。また、現在のネットワーク状況に応じてアクティブにするプロファイルを決定したあとに、そのプロファイルをアクティブにします。NWAM プロファイルの実装は、NWAM の主要なコンポーネントです。

ネットワークプロファイルとは

ネットワークプロファイルとは、現在のネットワーク状況に応じて、ネットワークの構成方法および操作方法を決定するプロバティーを集めたものです。
NWAM構成の概要

NWAM構成は、次のプロファイルタイプおよび構成オブジェクトで構成されています。

- ネットワーク構成プロファイル (NCP)
- 場所プロファイル
- 外部ネットワーク修飾子 (ENM)
- 既知のWLAN

主要な2つのネットワークプロファイルタイプは、NCPと場所プロファイルです。NWAMによるネットワークの自動構成を有効にするには、常にシステムで必ず1つのNCPと1つの場所プロファイルをアクティブにする必要があります。

NCPでは、物理リンクやIPインタフェースなどの個々のコンポーネントの構成を含むローカルネットワークの構成を指定します。各NCPは、ネットワーク構成ユーニット (NCU)と呼ばれる個々の構成オブジェクトで構成されています。各NCUは物理リンクまたはインタフェースを表し、そのリンクまたはインタフェースの構成を定義するプロパティーで構成されています。ユーザー定義のNCPを構成するプロセスには、そのNCPに関するNCUの作成が伴います。詳細は、49ページの『NCUの説明』を参照してください。

場所プロファイルには、次のようなシステム全体のネットワーク構成情報が含まれます。

- 場所プロファイルがアクティブになる条件
- 使用するネームサービス
- ドメイン名
- 一連のIPフィルタ規則
- IPsecポリシー

詳細は、50ページの『場所プロファイルの説明』を参照してください。

ENMは、ネットワーク構成を作成および変更できる外部アプリケーション用のNWAMプロファイルです。ENMの作成時に指定する条件で、これらの外部アプリケーションがアクティブまたは非アクティブになるようにNWAMを構成できます。

既知のWLANは、以前に接続した既知の無線ネットワークのリストを保持するために使用されるNWAMプロファイルです。詳細は、51ページの『ENMの説明』およびs2ページの『既知のWLANについて』を参照してください。

NCPの説明

NCPは、システムのネットワーク構成を定義します。NCPを構成するNCUには、起動されるインタフェース(複数可)、インタフェースが起動される条件、インタフェースのIPアドレスを取得する方法などの、さまざまなネットワークリンクおよびインタフェースの構成方法を指定します。NCPには、Automaticとユーザーや定義の2種類があります。Automatic NCPは、NWAMで自動的に作成されるシステム定義の
プロファイルです。このプロファイルを作成、変更、または削除することはできません。ユーザー定義のNCPは、特定のネットワーク構成の要件を満たすために作成するプロファイルです。ユーザー定義のNCPは、ユーザーによる変更および削除が可能です。

Automatic NCPは、現在システムに存在するすべてのリンクおよびインタフェースを表します。ネットワークデバイスの追加または削除されると、Automatic NCPの内容が変更されます。ただし、Automatic NCPに関連付けられた構成の設定は編集できません。Automatic NCPは、システムのIPアドレスの取得を可能にするDHCPおよびアドレスの自動構成を利用するプロファイルにアクセスできるように作成されています。このプロファイルには、無線リンクよりも有線リンクが優先されるリンク選択ポリシーも実装されています。代替のIP構成ポリシー、または代替のリンク選択ポリシーの指定が必要な場合は、追加のユーザー定義のNCPをシステムに作成します。

NCUの説明

NCUは、NCPを構成する個々の構成オブジェクトです。NCUは、システムに存在する個々の物理リンクおよびインタフェースを表します。ユーザー定義のNCPを構成するプロセスには、各リンクおよびインタフェースを構成する方法および条件を指定するNCUの作成が伴います。

NCUには次の2種類があります。

- リンク NCU
  リンク NCU（物理デバイスなど）は、開放型システム間相互接続（OSI）モデルの第2層エンティティです。

- インタフェース NCU
  インタフェース NCU（特にIPインタフェース）は、OSIモデルの第3層エンティティです。

リンク NCUはデータリンクを表します。データリンクには、次のようなさまざまなクラスがあります。

- 物理リンク（EthernetまたはWiFi）
- トンネル
- 集約
- 仮想ローカルエリアネットワーク（VLAN）
- 仮想ネットワークインタフェースカード（VNIC）
NWAM構の概要

注 - 現在のNWAM実装では、物理リンク（EthernetとWiFi）の基本ネットワーク構成のみがサポートされています。VNICやブリッジングなどのいくつかの高度なネットワーク技術は、NWAMでは積極的にサポートされていません。NWAMの構成管理を無効にしてもネットワーク上に構成できます。

ただし、IPネットワークマルチパス（IPMP）を使用するにシステムを構成する場合は、NWAMの構成管理を使用できません。従来のネットワーク構成を使用する必要があります。手順については、112ページの「自動ネットワーク構成モードから手動ネットワーク構成モードに切り替える方法」を参照してください。

Automatic NCPとユーザー定義のNCPの説明

Automatic NCPは、システムに存在する物理リンクごとに、1つのリンクNCUと1つのインタフェースNCUで構成されるシステム定義のプロファイルです。このNCPのNCUアクティブ化ポリシーでは、無線リンクよりも有線リンクの接続が優先され、有効な各リンクでIPv4とIPv6の両方がplumbされます。IPv4アドレスを取得するには、DHCPが使用されます。IPv6アドレスを取得するには、ステートレス自動構成およびDHCPが使用されます。Automatic NCPは、新しいリンクがシステムに挿入またはシステムから削除されると動的に変更されます。挿入または削除されたリンクに対応するすべてのNCUも、同時に追加または削除されます。プロファイルは、nwamdドーモンによって自動的に更新されます。

ユーザー定義のNCPは、ユーザーによって作成および管理されます。指定されたプロファイルからNCUを明示的に追加および削除する必要があります。現在システムに存在するリンクに関連しないNCUを作成できます。システムに存在するリンクに関連しないNCUを削除することもできます。さらに、ユーザー定義のNCPのポリシーを決定することもできます。たとえば、特定の時間にシステムで複数のリンクおよびインタフェースが有効になることを許可したり、NCUと静的IPアドレス間にさまざまな依存関係を指定したりできます。

ユーザー定義のNCPを作成して、このNCPにNCUを追加および削除するためのステップ・バイステップの手順については、80ページの「NCPを作成する」を参照してください。

場所プロファイルの説明

場所プロファイルでは、基本的なIP接続が確立されたあとで、追加ネットワークの詳細が提供されます。場所には、システム全体のレベルでのネットワーク構成に関連するプロパティセットで構成されるネットワーク構成情報が含まれています。
場所プロファイルは、ネームサービスやファイアウォールの設定などの特定の
ネットワーク構成情報で構成されています（必要に応じて、同時に適用されま
す）。また、場所は必ずしも物理的な場所に対応しているわけではないため、さまざまな
ネットワーク要件を満たすように複数の場所プロファイルを設定できます。た
とえば、会社のインターネットに接続するために、1つの場所を使用できます。オ
フィスに配置されている無線アクセスポイントを使用してパブリックイン
ターネットに接続するときに、もう1つの場所を使用できます。

デフォルトでは、2つの場所プロファイルがシステムで事前定義されています。

- NoNet
  場所「NoNet」には、非常に具体的なアクティブ化条件があります。このプロ
ファイルは、IPアドレスが割り当てられているローカルインタフェースがない場
合に、NWAMによってスタンドアロンシステムに適用されます。場
所「NoNet」は、システムではじめてアクティブにしたあとも変更できます。こ
の場所のデフォルト設定を変元する場合に備えて、元の場所「NoNet」の読み取
り専用コピーがシステムに格納されます。

- Automatic
  場所「Automatic」は、使用可能なネットワークが存在するが、その他の場所プロ
ファイルが優先される場合にアクティブになります。場所「Automatic」は、シ
ステムではじめてアクティブにしたあとに変更できます。この場所のデフォルト
設定を変元する場合に備えて、元の場所「Automatic」の読み取り専用コピーがシ
ステムに格納されます。

注 - 場所「Automatic」とAutomatic NCP を混同しないでください。場
所「Automatic」は、システムの初期ネットワーク構成が実行されたあとに、シス
テム全体のネットワークプロパティを定義する場所プロファイルの種類で
す。Automatic NCP は、システムのリンクおよびインタフェースのネットワーク
構成を指定します。

ユーザー定義の場所は、システム全体のネットワーク構成に指定される値で作成す
るプロファイルです。ユーザー定義の場所はユーザーが設定した値で構成されるの
に対して、システム定義の場所には事前定義された値があるという点を除い
て、ユーザー定義の場所はシステム定義の場所と同じです。

ユーザー定義の場所の作成の詳細は、88 ページの「場所プロファイルを作成す
る」を参照してください。

ENM の説明

ENM は、NWAM 外部にあるアプリケーションに関連するプロファイルです。これ
らのアプリケーションでは、ネットワーク構成の作成および変更が可能です。ENM
は、NCP または場所プロファイル以外のカスタマイズ配に基づきネットワーク構成を作
NWAM構成の概要

成および削除する手段として、NWAMの設計に含まれています。ENMは、有効時または無効時ネットワーク構成を直接変更するサービスまたはアプリケーションとして定義することもできます。指定された条件でENMがアクティブまたは非アクティブになるように、NWAMを構成できます。常にシステムで各プロファイルタイプの1つをアクティブにするNCPまたは場所プロファイルとは異なり、同時に複数のENMがシステムでアクティブになる可能性があります。常にシステムでアクティブなENMは、特にシステムで有効になるNCPまたは場所プロファイルに必ずしも依存しません。

ENMを作成できる外部アプリケーションおよびサービスはいくつか存在しますが、典型的な例としてVPNアプリケーションが挙げられます。システムにVPNをインストールして構成したあとに、指定された条件でアプリケーションが自動的にアクティブおよび非アクティブになるように、ENMを作成できます。

注-重要な点として、NWAMには、システムのネットワーク構成を直接変更できる外部アプリケーションについての自動的学習機能がないことを理解しておく必要があります。VPNアプリケーション、または任意の外部アプリケーションやサービスのアクティブ化または非アクティブ化を管理するには、まずアプリケーションをインストールしてから、CLIとNWAM GUIのどちらかを使用して、そのアプリケーション用のENMを作成する必要があります。

ENMで実行されるネットワーク構成に関する情報がNWAMによって格納または追跡される方法は、NCPまたは場所プロファイルに関する情報が格納される方法とまったく同様ではありません。ただし、NWAMには、外部で開始されたネットワーク構成を記述し、ENMによってシステム行われた構成の変更に基づいて、場所プロファイルをアクティブにする必要があるかどうかを再評価したあとに、その場所をアクティブにする機能があります。たとえば、特定のIPアドレスが使用中のときに、条件付きでアクティブ化される場所に切り替えることがありま
す。いつでもsvc:/network/physical：defaultサービスが再起動されると、アクティブなNCPで指定されたネットワーク構成が回復します。同様にENMが再起動されると、プロセス中にネットワーク構成が破棄され、再作成される可能性があります。

ENMのプロバティーの作成および変更については、93ページの「ENMプロファイルを作成する」を参照してください。

既知のWLANについて

既知のWLANは、システムに認識されている無線ネットワークを管理するためにNWAMで使用される機能オブジェクトです。NWAMでは、これらの既知の無線ネットワークのグローバルなリストが保持されます。その後、この情報は、NWAMが使用可能な無線ネットワークへの接続を試みる順序を決定する際に使用されます。「既知のWLAN」リストに存在する無線ネットワークが使用可能な場合
合、NWAMは自動的にそのネットワークに接続します。既知の無線ネットワークを2つ以上使用可能な場合、NWAMは優先度がもっとも高い（番号がもっとも小さい）無線ネットワークへの接続を試みます。NWAMが接続する新しい無線ネットワークは自動的に「既知のWLAN」リストの一番上に追加され、現在優先度がもっとも高い無線ネットワークになります。

既知のWLANは、符号のない整数で割り当てられた優先順位で選択されます。「既知のWLAN」リストでは、番号が小さいほど優先度が高いことを示します。はじめて無線ネットワークに接続するときに、NWAMによって自動的にそのWLANがリストに追加されます。新規WLANが追加されると、このリストで優先度がもっとも高いとみなされます。NWAMのデフォルト動作では、以前に接続されたWLANよりも最近接続されたWLANの方が優先されます。既知のWLANが同じ優先度を共有することは決してありません。既存のWLANと同じ優先度を持つ新規WLANがリストに追加されると、既存のエントリが低い優先度値にシフトします。その結果、リスト内のそのすべてのWLANの優先度値が動的に低い優先度値にシフトします。

既知のWLANに、1つ以上のキー名を関連付けることもできます。キー名があれば、dladm create-secobj コマンドを使用して独自のキーを作成できます。その後、既知のWLAN keyname ブロバティーにセキュリティ保護されたオブジェクト名を追加すると、これらのキーをWLANに関連付けることができます。詳細は、dladm(1M)のマニュアルページを参照してください。

NWAM コマンド行ユーティリティーを使用したWLAN管理の詳細は、121ページの「無線スキャンを実行して、使用可能な無線ネットワークに接続する」を参照してください。

**NWAM 構成データ**

システムには事実上、NWAM プロファイルリポジトリ (/etc/nwamディレクトリに格納される) と、従来の構成リポジトリ (/etc/ipadm/ipadm.conf と /etc/dladm/datalink.conf ファイル、およびネットワークサービスに関連付けられたその他の構成ファイルが含まれる) の2つの中リポジトリが存在します。

NWAMはネットワーク構成を管理するときに、主に独自のリポジトリから動作します。/etc/ipadm/ipadm.conf ファイルに格納されたインタフェース構成は無視されます。NWAMは、NCP データに基づいて物理リンクおよびインタフェースを直接構成します。

場所プロファイのデータは、NWAM プロファイルリポジトリから読み取られます。場所がアクティブになると、ほとんどの場合、適切なSMFサービスプロパティーを設定し、構成の変更が適用されるように対応するサービスを再起動することによって、この構成が実行中のシステムに適用されます。この操作では、これらのサービスプロパティーの既存の値が上書きされます。
NWAM構成データ

NWAMでは、場所プロファイルを適用するプロセス中にレガシー構成データが上書きされるため、起動時に、上書きされる可能性のある構成が保存されます。その後、NWAMはシャットダウン時にその構成を復元します。NWAM操作の一部として適用可能な場所ではありませんが、このデータはレガシー場所のデータと呼ばれています。

NWAMリポジトリには、次に示すシステム定義およびユーザー定義のネットワークプロファイールのプロパティ値が格納されます。

- NCP – Automatic NCP およびユーザーの定義の NCP の値が含まれます。
- NCU – リンク NCU とインタフェース NCU の両方の値が含まれます。
- 場所 – 3 種類のシステム定義の場所の値、およびユーザー定義の場所の値が含まれます。
- ENM – アプリケーションに関する情報が含まれます。
- 既知の WLAN – 自動的に接続される可能性のある無線ネットワークに関する情報が含まれます。

各 NCP の構成データは、ファイルとして/etc/nwam ディレクトリに ncp name という形式を使用して収納されます。各 NCU を表すエントリを含む 1 つのファイルが NCP ごとに存在します。たとえば、Automatic NCP のファイルは ncp-Automatic.conf という名前です。すべての NCP ファイルは、/etc/nwam ディレクトリに格納されます。

場所のプロパティは、/etc/nwam/loc.conf ファイルに格納されます。

ENM のプロパティは、/etc/nwam/enm.conf ファイルに格納されます。既知のWLANは、/etc/nwam/known-wlan.conf ファイルに格納されます。このファイル形式は、/etc/dladm/datalink.conf ファイルのファイル形式と似ています。

注 - NWAM プロファイルリポジトリのファイルを直接編集しても、ネットワークプロファイルを変更することは不可能ですが、プロファイルを変更するための適切な方法は、netcfg コマンドまたは NWAM GUI 構成パネルを使用することです。ファイル形式およびファイルの使用法は、今後のリリースで変更される可能性があります。100ページの「プロファイルのプロパティ値を設定および変更する」を参照してください。

NCUプロパティの値

NCUはNCPの個々の構成オブジェクトであり、システム上の個々のリンクおよびインタフェースを表します。両方のNCUタイプ(リンクとインタフェース)の一般プロパティ、および各NCUタイプに固有のプロパティは、NWAM プロファイルリポジトリに格納されます。type、class、およびparentプロパティは、NCUの作成時に設定され、あとから変更することはできません。また、enabledプロパ
ディーも直接変更できません。このプロパティは、netadmコマンドを使用して
NCUを有効または無効にすることによって、間接的に変更されます。

Automatic NCP は、システムで検出された物理リンクごとに1つのリンク NCU
と、各リンクで plumb された1つのインタフェース NCU で構成されます。Automatic
NCP は、追加の物理リンクの挿入時に動的に変更されます。新しいリンクが挿入さ
れると、新しいリンクごとに、リンク NCU および対応するインタフェース NCU が
作成されます。次の表では、Automatic NCP を構成する各 NCU に割り当てられる値
を定義します。

注 - この表に示すプロパティは、Automatic NCP の NCU プロパティを参照すると
きに表示される順序で一覧表示されています。各 NCU タイプには、特定の値が適用
されます。

<table>
<thead>
<tr>
<th>プロパティ</th>
<th>リンク NCU の値</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>link</td>
</tr>
<tr>
<td>class</td>
<td>phys</td>
</tr>
<tr>
<td>parent</td>
<td>Automatic</td>
</tr>
<tr>
<td>enabled</td>
<td>true</td>
</tr>
<tr>
<td>activation-mode</td>
<td>prioritized</td>
</tr>
<tr>
<td>priority-group</td>
<td>0 (802.3 リンク) または 1 (802.11 リンク)</td>
</tr>
<tr>
<td>priority-group-mode</td>
<td>shared (802.3 リンク) または exclusive (802.11 リンク)</td>
</tr>
<tr>
<td>mac-address</td>
<td>ハードウェアによる割り当て</td>
</tr>
<tr>
<td>autopush</td>
<td>なし</td>
</tr>
<tr>
<td>MTU</td>
<td>なし</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>プロパティ</th>
<th>インタフェース NCU の値</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>interface</td>
</tr>
<tr>
<td>class</td>
<td>IP</td>
</tr>
<tr>
<td>parent</td>
<td>Automatic</td>
</tr>
<tr>
<td>enabled</td>
<td>true</td>
</tr>
<tr>
<td>ip-version</td>
<td>ipv4, ipv6</td>
</tr>
</tbody>
</table>
システム定義の場所のプロパティ値
次の表には、システム定義のプロファイアルである場所「Automatic」のデフォルトプロパティ値を示します。これらの値は、activation-mode と enabled プロパティを除いて変更可能です。1つ以上のインタフェースがアクティブであり、その他の場所プロファイアルが優先されない場合は、常にシステムで場所「Automatic」がアクティブになります。

<table>
<thead>
<tr>
<th>プロパティ</th>
<th>値</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Automatic</td>
</tr>
<tr>
<td>activation-mode</td>
<td>system</td>
</tr>
<tr>
<td>enabled</td>
<td>system必要に応じて変更される</td>
</tr>
<tr>
<td>conditions</td>
<td>なし</td>
</tr>
<tr>
<td>default-domain</td>
<td>なし</td>
</tr>
<tr>
<td>nameservices</td>
<td>dns</td>
</tr>
<tr>
<td>nameservices-config-file</td>
<td>/etc/nsswitch.dns</td>
</tr>
<tr>
<td>dns-nameservice-configsrc</td>
<td>dhcp</td>
</tr>
<tr>
<td>dns-nameservice-domain</td>
<td>なし</td>
</tr>
<tr>
<td>dns-nameservice-servers</td>
<td>なし</td>
</tr>
<tr>
<td>dns-nameservice-search</td>
<td>なし</td>
</tr>
<tr>
<td>nis-nameservice-configsrc</td>
<td>なし</td>
</tr>
<tr>
<td>nis-nameservice-servers</td>
<td>なし</td>
</tr>
<tr>
<td>ldap-nameservice-configsrc</td>
<td>なし</td>
</tr>
<tr>
<td>ldap-nameservice-servers</td>
<td>なし</td>
</tr>
</tbody>
</table>
次の表には、場所「NoNet」の定義済みプロバティーを示します。これらの値は、activation-mode と enabled プロバティーを除いて変更可能であることに注意してください。アクティブなインタフェースがない場合は、常にシステムで場所 NoNet が有効になります。

<table>
<thead>
<tr>
<th>プロバティー</th>
<th>値</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>NoNet</td>
</tr>
<tr>
<td>activation-mode</td>
<td>system</td>
</tr>
<tr>
<td>enabled</td>
<td>system必要に応じて変更される</td>
</tr>
<tr>
<td>conditions</td>
<td>なし</td>
</tr>
<tr>
<td>default-domain</td>
<td>なし</td>
</tr>
<tr>
<td>nameservices</td>
<td>files</td>
</tr>
<tr>
<td>nameservices-config-file</td>
<td>/etc/nsswitch.files</td>
</tr>
<tr>
<td>dns-nameservice-configsrc</td>
<td>なし</td>
</tr>
<tr>
<td>dns-nameservice-domain</td>
<td>なし</td>
</tr>
<tr>
<td>dns-nameservice-servers</td>
<td>なし</td>
</tr>
<tr>
<td>dns-nameservice-search</td>
<td>なし</td>
</tr>
<tr>
<td>nis-nameservice-configsrc</td>
<td>なし</td>
</tr>
<tr>
<td>nis-nameservice-servers</td>
<td>なし</td>
</tr>
<tr>
<td>ldap-nameservice-configsrc</td>
<td>なし</td>
</tr>
<tr>
<td>ldap-nameservice-servers</td>
<td>なし</td>
</tr>
</tbody>
</table>
NWAM プロファイルをアクティブにする方法

表3-4 場所「NoNet」のプロバティー (続き)

<table>
<thead>
<tr>
<th>プロバティー</th>
<th>値</th>
</tr>
</thead>
<tbody>
<tr>
<td>nfsv4-domain</td>
<td>なし</td>
</tr>
<tr>
<td>ipfilter-config-file</td>
<td>/etc/nwam/loc/NoNet/ipf.conf (DHCP アドレスの割り当てなどのネットワーク構成を実行するために NWAM で必要最小量のネットワークトラフィックを除く、ループバック以外のすべてのトラフィックをブロックする IP 振り分け規則で構成される)</td>
</tr>
<tr>
<td>ipfilter-v6-config-file</td>
<td>/etc/nwam/loc/NoNet/ipf6.conf (ipfilter-config-file の説明のとおり、IP 振り分け規則で構成される)</td>
</tr>
<tr>
<td>ipnat-config-file</td>
<td>なし</td>
</tr>
<tr>
<td>ippool-config-file</td>
<td>なし</td>
</tr>
<tr>
<td>ike-config-file</td>
<td>なし</td>
</tr>
<tr>
<td>ipsecpolicy-config-file</td>
<td>なし</td>
</tr>
</tbody>
</table>

場所のプロバティー (ユーザー定義の場所を構成するプロバティーを含む) の詳細は、netcfg(1M) のマニュアルページを参照してください。

NWAM プロファイルをアクティブにする方法

NCP、場所プロファイル、およびENM には、activation-mode プロバティーが存在します。許容される値は、プロファイルタイプごとに異なります。さらに、activation-mode プロバティーを検証する方法も、各プロファイルがアクティブになる条件と同様に、プロファイルごとに異なります。

システム定義の場所 (Automatic と NoNet) では、activation-mode プロバティー値が system に設定されています。つまり、場所はシステムでのみアクティブにでき、さらにシステムで事前に定められた条件がその場所に対して適用でなければなりません。

ユーザー定義の場所では、activation-mode と conditions プロバティーを manual、conditional-any、または conditional-all に設定できます。詳細は、61 ページの「場所のアクティブ化の選択条件」を参照してください。

netadm コマンドまたは NWAM GUI を使用すると、場所プロファイルを手動で有効にすることができます。場所を明示的に有効にしない場合は、NWAM デーモン nwamd によって、条件付きでアクティブ化およびシステムでアクティブ化されたすべての場所プロファイルのアクティブ化規則がチェックされたあとに、現在のネットワーク環境に最適な場所が選択されます。
NWAM では、場所の選択で「最適な一致」を決定するアルゴリズムが使用されます。場所に適した一致がない場合は、場所「Automatic」がアクティブになります。ネットワーク環境を変更すると、nwamd デーモンが場所の選択を継続的に再評価して、最適な一致を決定します。ただし、netadm コマンドを使用して場所プロファイル（手動でアクティブ化した場所または条件付きでアクティブ化した場所）を明示的に有効にする場合は、その場所を明示的に無効にするか、別の場所を有効にするまで、場所はアクティブなままです。この状況では、より適切な一致があるかどうかに関係なく、ネットワーク環境を変更しても場所プロファイルは変更されません。現在の場所を明示的に指定することにより、実質的には最良的一致が決定されます。プロファイルをアクティブまたは非アクティブにする手順については、118 ページの「プロファイルをアクティブまたは非アクティブにする」を参照してください。

### NCP アクティブ化ポリシー

NWAM では、NCU がアクティブな場合には、NCP ポリシーを指定できます。NCP ポリシーは、NCU ごとに指定可能なプロパティーおよび条件を使用することで強制されます。指定されるポリシーの例として、「無線接続よりも有線接続を優先する」や「一度に1つのインタフェースをアクティブにする」などがあります。NCP をアクティブにする方法および条件は、NCU タイプごとに設定されたプロパティーで定義されます。

注 - 常にインタフェース NCU は、ベースとなるリンク NCU に関連付ける必要があります。各インタフェース NCU は、関連付けられたリンク NCU をアクティブにするアクティブになります。netadm コマンドを使用すると、NCU のデフォルト動作を上書きすることができます。ただし、ベースとなるリンク NCU との依存性は削除できません。たとえば、関連付けられたリンク NCU を有効にせずにインタフェース NCU を有効にする場合は、そのインタフェースのベースとなる NCU がアクティブになるまで、インタフェースは実際にオンラインになりません。

### NCP ポリシーの例

次の例では、使用可能な有線リンクがすべてアクティブであること、および使用可能な有線接続がない場合にのみ無線接続を使用するように NCP ポリシーで指定するために NCU プロパティーが設定されます。

すべての物理リンク用:

- NCU タイプ: link
- NCU クラス: phys
- activation-mode: prioritized
- priority-group: 0(有線), 1(無線)
- priority-mode: shared(有線), exclusive(無線)
NWAMプロファイルをアクティブにする方法

次の例では、常にシステムでアクティブなリンクは1つのみであること、および無線接続よりも有線接続が優先されることを指定したNCPポリシーに従って、NCPプロパティが設定されています。

すべての物理リンク用:
- NCUタイプ:link
- NCUクラス:phys
- activation-mode:prioritized
- priority-group:0(有線), 1(無線)
- priority-mode:exclusive

NCUのアクティブ化プロパティ
ネットワーク接続をアクティブにする方法は、リンクNCUプロパティで設定されます。NCPのアクティブ化ポリシーを定義する際には、次のプロパティが使用されます。

- activation-modeプロパティ
  このプロパティには、manualとprioritizedのどちらかを設定できます。
  - manual - NCUのアクティブ化が管理者によって管理されます。NWAM CLI またはGUIを使用して、NCU をアクティブまたは非アクティブにすることができます。NCU のactivation-modeがmanualに設定されている場合は、priority-groupとpriority-mode NCUプロパティに設定された値はどちらも無視されます。
  - prioritized - 指定されたNCUのpriority-groupとpriority-modeプロパティで設定された値に従って、NCUがアクティブになります。優先順位が付けられたNCUでは、有効なプロパティは常にtrueです。

優先順位が付けられたアクティブ化によって、リンクのグループを同時にアクティブにすることができます。この起動モードでは、1つ以上のリンクを他のリンクよりも優先させることもできます。priority-groupプロパティでは、指定されたリンクに数値の優先度が割り当てられます。優先度が同じリンクはすべて、1つのグループとして検査されます。priority-modeプロパティは、アクティブにするグループで使用可能になる可能性のある、または使用可能にする必要があるグループメンバーの数を定義します。

- enabledプロパティ(activation-modeはmanualに設定される)
  このプロパティに指定可能な値はtrueまたはfalseです。このプロパティの値は設定できません。より正確には、この値は手動で有効にしたNCUの現在の状態を反映しており、netadmコマンドまたはNWAM GUIを使用すれば変更できます。

- priority-groupプロパティ(activation-modeはprioritizedに設定される)
  値は数値です。ゼロ(0)は優先度がもっとも高いことを示します。負の値は無効です。

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
使用可能なすべてのpriority-groupの中で、priority-groupがもっとも高いNCUのみがアクティブになります。優先度が同じNCUが複数使用可能である場合、アクティブ化動作はpriority-modeプロバティーで定義されます。優先度番号は絶対値ではありません。NCPリポジトリが更新されると、値が変更される可能性があります。

注 - 優先度の順序は厳格に適用されます。

- priority-modeプロバティー(activation-modeはprioritizedに設定される)
  このプロバティーは、priority-groupプロバティーの値が指定されている場合に設定されます。

  このプロバティーの値は次のとおりです。
  - exclusive - 常にpriority-group内で1つのNCUのみをアクティブにできることを指定します。NWAMでは、優先グループ内で1番目に使用可能なNCUがアクティブになり、その他のNCUは無視されます。
  - shared - 優先グループ内の複数のNCUを同時にアクティブにできることを指定します。優先グループ内で使用可能な任意のNCUがアクティブになります。
  - all - 優先グループが使用可能であり、したがってアクティブにできるとみなされるには、優先グループ内のすべてのNCUが使用可能でなければならないことを指定します。

場所のアクティブ化の選択条件

各場所プロファイルには、アクティブ化の条件を定義したプロバティーが存在します。これらのプロバティーは、場所がアクティブになる条件に関する情報を指定します。NWAMでは、構成されたすべての場所に対する選択条件を継続的に再評価し、そのたびに、現在のネットワーク環境に最適な条件を持つ場所を判定します。現在のネットワーク環境に変更が発生した結果、より条件に合致するプロファイルがある場合、NWAMは現在の場所プロファイルを非アクティブにして、新しい環境により適した場所プロファイルをアクティブにします。

場所をアクティブにするタイミングと方法の選択条件は、次のプロバティーで指定されます。

- activation-mode
- conditions
activation-mode プロパティは、次の値のいずれかに設定されます。

- manual
- conditional-any
- conditional-all
- system

注 - activation-mode プロパティの system 値は、システムで提供される場所 「Automatic」および場所「NoNet」にのみ割り当てることができます。system 値は、これらの場所がアクティブになる条件がシステムで決定されることを示します。

activation-mode プロパティが conditional-any または conditional-all に設定されている場合は、conditions プロパティにユーザーディスプレイの条件式(複数可)が含まれます。各形式含まれる条件には、ブール型の値を割り当てることができます（「ncu ip:net0 is-not active」など)。

activation-mode プロパティが conditional-any に設定されている場合は、条件のいずれかに該当すれば条件を満たすことになります。

activation-mode プロパティが conditional-all に設定されている場合は、条件のすべてに該当しなければ条件を満たすことになりません。次の表では、条件文字列を作成する際に使用可能な条件および演算を定義します。

表3-5 条件文字列を作成するための条件と演算

<table>
<thead>
<tr>
<th>オブジェクト型/属性</th>
<th>条件</th>
<th>オブジェクト</th>
</tr>
</thead>
<tbody>
<tr>
<td>ncu, enm, loc</td>
<td>is/is-not active</td>
<td>名前</td>
</tr>
<tr>
<td>essid</td>
<td>is/is-not contains/does-not-contain</td>
<td>名前文字列</td>
</tr>
<tr>
<td>bssid</td>
<td>is/is-not</td>
<td>bssid文字列</td>
</tr>
<tr>
<td>ip-address</td>
<td>is/is-not</td>
<td>IPv4 または IPv6 アドレス</td>
</tr>
<tr>
<td>ip-address</td>
<td>is-in-range/is-not-in-range</td>
<td>IPv4 または IPv6 アドレスとネットマスク/接頭辞の長さ</td>
</tr>
<tr>
<td>advertised-domain</td>
<td>is/is-not contains/does-not-contain</td>
<td>名前文字列</td>
</tr>
<tr>
<td>system-domain</td>
<td>is/is-not contains/does-not-contain</td>
<td>名前文字列</td>
</tr>
</tbody>
</table>
注-ssid プロバティーは、無線 LAN (WLAN) のネットワーク名である Extended Server Set Identifier (ESSID) を表します。bssid プロバティーは、特定の無線アクセスポイント (WAP) または任意のアクセスポイント(AP) の MAC アドレスである Basic Service Set Identifier (BSSID) を表します。

advertised-domain 属性と system-domain 属性との区別に注意してください。通知されたドメインは、外部通信によって検出されます。たとえば、DNSdomainやNISdomainドメイン名は、DHCP サーバーによって通知されます。この属性は、場所の条件付きアクティブ化に役立ちます。たとえば、通知されたドメインがmycompany.comである場合、workという場所をアクティブ化します。system-domain属性は、現在システムに割り当てられているドメインです。また、domainnameコマンドで返される値です。この属性は、場所がアクティブ化されており、その場所のドメインに対してシステムが構成されてはじめてtrueになるので、ENMの条件付きのアクティブ化に役立ちます。詳細は、domainname(1M)のマニュアルページを参照してください。

場所のプロバティーの詳細は、50ページの「場所プロファイルの説明」を参照してください。

netcfgを使用してプロファイルを構成する

ネットワークプロファイルのプロバティーおよび値を構成するには、netcfg(1M)のマニュアルページで説明されているnetcfgコマンドを使用します。

netcfgコマンドを使用して、次のタスクを実行できます。

- ユーザー定義のプロファイルを作成または削除します。

注-システム定義のプロファイルを作成または削除することはできません。

- システムに存在するプロファイルおよびそのプロバティー値をすべて一覧表示します。
- 指定されたプロファイルのプロバティー値およびリソースをすべて一覧表示します。
- プロファイルに関連付けられた各プロバティーを表示します。
- 指定されたプロファイルのプロバティーの1つまたはすべてを設定または変更します。
- ユーザー定義のプロファイルの現在の構成を標準出力またはファイルへエクスポートします。
netcfgを使用してプロファイルを構成する

注-システム定義のプロファイルはエクスポートできません。

- プロファイルにわたる変更を削除して、そのプロファイルの以前の構成に戻します。
- プロファイルの構成が有効であるかどうかを検証します。

netcfgユーザーアンタフェースは、対話型モード、コマンド行モード、またはコマンドファイルモードで使用できます。netcfgコマンドは階層構造であるため、対話型モードで使用した方が理解が容易です。

netcfgコマンドでは、スコープの概念が使用されます。このコマンドを対話形式で使用する場合、スコープは常にプロファイルタイプおよび実行中のタスクに依存します。端末ウィンドウでnetcfgコマンドを入力すると、プロンプトがグローバルスコープに表示されます。

ここから、selectまたはcreateサブコマンドを使用して、次の最上位プロファイルを表示、変更、または作成できます。

- NCP
- 場所
- ENM
- 既知のWLAN

プロファイルを作成または選択する前に、netcfg対話型プロンプトが次の形式で表示されます。

netcfg>

プロファイルを作成または選択したあとは、netcfg対話型プロンプトが次のように表示されます。

netcfg:profile-type:profile-name>

注-コマンド行モードでは、完全なコマンドを単一行で入力する必要があります。netcfgコマンドをコマンド行モードで使用することによって、選択したプロファイルに行った変更は、コマンド入力が完了するとすぐに永続リポジトリに確定されます。

netcfgコマンドを使用するためのステップ・バイ・ステップの手順については、第4章「NWAMプロファイルの構成(タスク)」を参照してください。netcfgコマンドの使用については、netcfg(1M)のマニュアルページを参照してください。
**netcfg対話型モード**

`netcfg`対話型モードでは作業中に更に詳細な設定を変更したり、コマンドプロンプトが場所プロファイルおよびENMのプロファイルスコープに表示されます。例:

```
netcfg> select loc foo
netcfg:loc:foo>
```

NCPが選択された場合は、コマンドプロンプトがNCPスコープに表示されます。NCPスコープから、NCUを選択または作成できません。NCUを選択または作成するとき、選択したNCUのプロファイルスコーププロンプトになります。このスコープでは、次の例で示すように、現在選択されているプロファイルに関連付けられたすべてのプロパティを表示および設定できます。ここでは、まず`ncp User`が選択されたあとに、NCUがNCPスコープで作成されました。この操作により、新たに作成されたNCUのプロファイルスコープになります。このスコープでは、NCUのプロパティを表示または設定できます。

```
netcfg> select ncp User
netcfg:ncp:User> create ncu phys net2
Created ncu 'net2'. Walking properties ...
activation-mode (manual) [manual|prioritized]>
```

どのスコープでも、コマンドプロンプトは現在選択されているプロファイルを示します。このスコープでプロファイルを行った変更は、確定できます。つまり、変更は永続リポジトリに保存されます。変更は、スコープの終了時に暗黙的に確定されます。選択したプロファイル行った変更を確定しない場合は、そのプロファイルを最後に確定したときの状態に戻すことができます。この操作を行うと、そのレベルでプロファイル行った変更が元に戻ります。`revert`と`cancel`サブコマンドは同じように動作します。

**netcfgコマンド行モード**

コマンド行モードでは、選択したプロファイルまたはプロパティに影響を与えるサブコマンドは、選択したプロファイルまたはプロパティが存在する特定のスコープで実行する必要があります。たとえば、NCUのプロパティ値を取得するには、`get`サブコマンドをその特定のNCUのスコープで使用します。`netcfg`対話型モードでは、このコマンドで使用する構文は比較的明確です。ただし、コマンド行モードでは構文があまり明確でない場合があります。

たとえば、User NCPで`myncu`というNCUの属性である、プロパティ「foo」の値を取得するには、次の構文を使用します。

```
$ netcfg "select ncp User; select ncu ip myncu; get foo"
```
この例では、次の情報に注意してください。

- 各スコープはセミコロンで区切られます。
- `select`サブコマンドは、各スコープで（グローバルスコープとプロファイルスコープで1回ずつ）発行されます。
- `get`サブコマンドは、プロパティ「foo」が存在するスコープ内で使用されます。
- シェルでセミコロンが解釈されないようにするには、二重引用符が必要です。

### netcfgコマンドファイルモード

コマンドファイルモードでは、構成情報がファイルから取得されます。このファイルを作成するには、`export`サブコマンドを使用します。その後、構成を標準出力に出力するか、`-o`オプションで指定した出力ファイルに出力することができます。`export`サブコマンドは、対話形式でも使用できます。詳細は、66ページの「netcfgでサポートされるサブコマンド」を参照してください。

### netcfgでサポートされるサブコマンド

対話型モードおよびコマンド行モードでは、次のnetcfgサブコマンドがサポートされています。特定のサブコマンドでは、各スコープで異なるセマンティクスがあることに注意してください。サブコマンドを特定のモードで使用できない場合は、サブコマンドの説明に注記されています。

- `cancel`
  現在の変更を永続ストレージに確定せずに現在のプロファイル指定を終了してから、1つ上のレベルである前のスコープに進みます。
- `clear prop-name`
  指定したプロパティの値を消去します。
- `commit`
  永続ストレージに現在のプロファイルを確定します。確定されるのは構成が正確である必要があります。したがって、この操作では、`verify`も同様にプロファイルまたはオブジェクトで自動的に実行されます。`commit`操作は、`end`サブコマンドまたは`exit`サブコマンドを使用して現在のスコープを終了するときに、自動的に試行されます。
- `create [-t template] object-type [ class ] object-name`
  指定されたタイプと名前のインメモリープロファイルを作成します。`-t template`オプションは、新しいプロファイルが、`template`と同一であるべきことを指定します。ここで、`template`は同じタイプの既存のプロファイルの名前です。`-t`オプションを使用しない場合、新しいプロファイルはデフォルト値で作成されます。
netcfgを使用してプロファイルを構成する

- **destroy -a**
  すべてのユーザー定義のプロファイルをメモリーおよび永続ストレージから削除します。

- **destroy object-type [ class ] object-name**
  指定したユーザー定義のプロファイルをメモリーおよび永続ストレージから削除します。

注意-この操作は即座に実行され、確定する必要はありません。破棄したプロファイルを元に戻すことはできません。

- **end**
  現在のプロファイルを指定し、1つ上のレベルである前のスコープに進みます。編集操作を終了する前に、現在のプロファイルが検証および確定されます。verifyとcommitのどちらかの操作に失敗すると、エラーメッセージが表示されます。その後、現在の変更を確定せずに操作を終了する機会が与えられます。また、現在のスコープに残って、プロファイルの編集を続行することもできます。

- **exit**
  netcfg対話型セッションを終了します。現在のセッションが終了する前に、現在のプロファイルが検証および確定されます。verifyとcommitのどちらかの操作に失敗すると、エラーメッセージが表示されます。その後、現在の変更を確定せずにセッションを終了する機会が与えられます。また、現在のスコープに残って、プロファイルの編集を続行することもできます。

- **export [ -d ] [- f output-file ] [ object-type [ class ] object-name ]**
  現在または指定したスコープの現在の構成を、標準出力または-fオプションで指定したファイルに出力します。-dオプションは、出力の最初の行としてdestroy -aサブコマンドを生成します。このサブコマンドを使用すると、コマンドファイルに適した形式で出力が生成されます。

注- Automatic NCP およびAutomatic、NoNet、レガシー場所などのシステム定義のプロファイルは、エクスポートできません。

- **get [ -V ] prop-name**
  指定したプロパティーの現在のインメモリー値を取得します。デフォルトでは、プロパティーの名前と値の両方が出力されます。-Vオプションが指定されている場合は、プロパティー値のみが出力されます。

- **help [ subcommand ]**
  一般ヘルプまたは特定の項目に関するヘルプを表示します。
netcfgを使用してプロファイルを構成する

- list [-a] [object-type [class] object-name]
  現在または指定したスコープで使用されるすべてのプロファイル、プロバ
  ティーと値のペア、およびリソースを一覧表すします。-aオプションが指定され
  ている場合は、現在の設定に基づいて、無視されるプロバティーを含むすべての
  プロバティーが一覧表すます。

- revert
  プロファイル行われれた現在の変更を削除してから、永続ストレージの値に戻
  いまます。

- select object-type [class] object-name
  指定されたオブジェクトを選択します。

- set prop-name= value
  指定したプロバティーの現在のインメモリ値を設定します。
  コマンド行モードで実行した場合、永続ストレージにも変更がただちに確定され
  ます。
  複数の値が含まれるプロバティーの区切り文字は、コンマ(,)です。指定された
  プロバティーの各値にコンマが含まれている場合は、前にバックラッシュ(\)
  を付ける必要があります。単一の値のみが含まれるプロバティー内のコンマ
  は、区切り文字は解釈されないため、前にバックラッシュを付ける必要はあ
  りません。

- verify
  現在のインメモリープロファイルまたはオブジェクトの構成が有効であるかどうかを検証します。

- walkprop [-a]
  現在のプロファイルに関連付けられたプロバティーを「調査」します。プロバ
  ティーごとに、名前と現在の値が表示されます。現在の値を変更できるブロンプ
  トが表示されます。以前に指定した値に基づいて、プロバティーが使用されない
  場合は、プロバティーが表示されません。たとえば、ipv4-addrsrcプロバ
  ティーがstaticに設定されている場合、-aオプションを指定しないかぎ
  り、ipv4-addrプロバティーは使用されず、調査や一覧表すも行われません。
  -aオプションが使用されると、指定したプロファイルまたはオブジェクトに対し
  て使用可能なすべてのプロバティーが繰り返されます。
  複数の値が含まれるプロバティーの区切り文字は、コンマ(,)です。指定された
  プロバティーの各値にコンマが含まれている場合は、前にバックラッシュ(\)
  を付ける必要があります。単一の値のみが含まれるプロバティー内のコンマ
  は、区切り文字は解釈されないため、前にバックラッシュを付ける必要はあ
  りません。
netadm コマンドを使用してプロファイルを管理する

プロファイル (NCP、場所、ENM、WLAN) および NCU (NCP を構成する個々の構成オブジェクト) のステータスを管理および取得するには、netadm コマンドを使用します。さらに、GUI が存在しない場合でも、netadm コマンドを使用すれば、NWAM デーモン (nwamd) と対話できます。netadm の詳細は、netadm(1M) のマニュアルページを参照してください。

次の netadm サブコマンドがサポートされています。

- `enable [-p profile-type] [-c ncu-class] profile-name`
  指定されたプロファイルを有効にします。プロファイル名が一致でない場合
  は、プロファイルタイプを指定する必要があります。プロファイルタイプが ncu
  であり、名前が一致でない（たとえば、同じ名前のリンクとインタフェース ncu の
  両方がある）場合、-c オプションを使用して NCU クラスを指定しないかぎり、両
  方の NCU が有効化されます。

  プロファイルタイプは、次のいずれかである必要があります。
  - ncp
  - ncu
  - loc
  - enm
  - wlan

  NCU クラスには、phys と ip のいずれかを指定する必要があります。

- `disable [-p profile-type] [-c ncu-class] profile-name`
  指定されたプロファイルを無効にします。プロファイル名が一致でない場合、無
  効にするプロファイルを識別するためにプロファイルタイプを指定する必要があります。
  プロファイルタイプが ncu であり、名前が一致でない（たとえば、同じ名
  前のリンクとインタフェース ncu の両方がある）場合、-c オプションを使用して
  NCU クラスを指定しないかぎり、両方の NCU が無効化されます。

  プロファイルタイプは、次のいずれかである必要があります。
  - ncp
  - ncu
  - loc

注 - このサブコマンドは、対話型モードで使用する場合にのみ有効です。

タスク関連の情報については、第 4 章「NWAM プロファイルの構成 (タスク) 」を参
照してください。
netadm コマンドを使用してプロファイルを管理する

- enm
- wlan

NCU クラスには、phys と ip のいずれかを指定する必要があります。

- list [-x] [-p profile-type] [-c ncu-class] [profile-name]

使用可能なプロファイルおよびその現在の状態をすべて一覧表示します。取り得る状態の値については、次の節で一覧表示されています。プロファイルを名前で指定した場合は、そのプロファイルの現在の状態のみが表示されます。プロファイル名が一意でない場合は、指定された名前のプロファイルがすべて一覧表示されます。また、特定のプロファイルを識別するために、プロファイルタイプまたは NCU クラス（あるいは両方）を含めることもできます。プロファイルタイプだけを指定すると、そのタイプのプロファイルがすべて一覧表示されます。

有効な NCP を一覧表示すると、その NCP を構成する NCU もすべて含まれます。

-x オプションが指定されている場合は、一覧表示される各プロファイルの状態に関する詳細な説明も出力に含まれます。

発生可能なプロファイル状態の値は、次のとおりです。

- disabled
  まだ有効化されていないプロファイルが手動でアクティブ化されたことを示します。

- offline
  まだ有効化されていないプロファイルが条件付きでアクティブ化またはシステムでアクティブ化されたことを示します。条件を満たしていないために、プロファイルがアクティブになる場合があります。また、より具体的な条件を満たさないプロファイルが代わりにアクティブになっているために、プロファイルがアクティブにならない場合もあります。この条件は、一度に1つずつ有効にする必要があるプロファイルタイプ（場合プロファイルなど）に適用されます。

- online
  すでに条件を満たし、正常に有効になっているプロファイルが条件付きでアクティブ化またはシステムでアクティブ化されたことを示します。または、手動でアクティブ化されたプロファイルが、ユーザーの要求で正常に有効化されたことを示す場合もあります。

- maintenance
  プロファイルの有効化が試行されたが、失敗したことを示します。

- initialized
  プロファイルが、操作がまだ行われていない有効な構成オブジェクトを表していることを示します。

- uninitialized
NWAM デーモンの概要

NWAM デーモンは、nwamd デーモンと netcfgd デーモンの 2つのデーモンが使用されます。ポリシーエンジンデーモンである nwamd は、複数の役割で動作することによってネットワークの自動構成を制御します。リポジトリデーモンである netcfgd は、ネットワーク構成リポジトリへのアクセスを制御します。

NWAM ポリシーエンジンデーモン (nwamd) の説明

nwamd デーモンは、次の役割を引き受けることによってネットワークの自動構成を制御します。

- イベントコレクタ
  この役割では、ルーティングソケットおよび sysevent の登録によって検出する必要のあるリンク関連のイベントを収集します。nwamd によるこのタスクの実行例として、NICがシステムにホットプラグされたことを示す EC_DEV_ADD sysevent をデーモンが取得することが挙げられます。このようなイベントはすべて、nwamd イベント構造にパッケージ化されてから、該当タスクを担当するイベント処理スレッドに送信されます。

- イベントハンドラ
  この役割では、対象のイベントに応答するためのイベントループスレッドを実行します。イベントハンドラは、NWAM サービスで管理されるさまざまなオブジェクトに関連付けられた状態マシンで動作します。イベントの処理中
SMFネットワークサービス

に、nwamdデーモンは、ネットワーク環境の変更を検出します。その結果として、1つまたは複数のプロファイルへの変更がトリガーされる場合があります。

イベントディスパッチャー
この役割では、このようなイベントへの関与を登録した外部コンシューマにイベントを送信します。イベントディスパッチの例として、使用可能なWLANに関する情報を含む無線スキャンイベントが挙げられます。これは、NWAM GUIの使用時に役立ちます。一方、GUIでは使用可能なオプションをユーザーに表示できます。

プロファイルマネージャー
nwamdデーモンによるこれらのプロファイルの管理には、次的情報に応じたネットワーク構成の適用が含まれます。
- アクティブにするリンクとインタフェース
- 接続されているネットワークの特性
- 有効なプロファイルに組み込まれた偶発性と依存性
- 受信される外部イベント

NWAMリポジトリデーモン (netcfgd) の説明
プロファイルデーモンであるnetcfgdは、ネットワーク構成リポジトリへのアクセスを制御および管理します。このデーモンは、svc:/network/netcfgd/default SMFサービスによって自動的に開始されます。このデーモンによって、情報をリポジトリから読み取り、リポジトリに書き込んだりする任意のアプリケーションに、次の承認が付与されます。

- solaris.network.autoconf.read
- solaris.network.autoconf.write

承認の詳細は、auth_attr(4)のマニュアルページを参照してください。セキュリティプロファイルの詳細は、prof_attr(4)のマニュアルページを参照してください。

netcfgdデーモンの詳細は、netcfgd(1M)のマニュアルページを参照してください。

SMFネットワークサービス

Oracle Solarisでは、ネットワーク構成が複数のSMFサービスで実装されています。
- svc:/network/loopback:default-IPv4およびIPv6のループバックインタフェースを作成します。
NWAMセキュリティーの概要

NWAMのセキュリティーは、次のコンポーネントが組み込まれるように設計されています。

- CLI (netcfgとnetadmコマンド)
- NWAM GUI
- NWAMプロファイルリポジトリデーモン (netcfgd)
- ポリシーエンジンデーモン (nwamd)
- NWAMライブラリ (libnwam)

netcfgdデーモンは、すべてのネットワーク構成情報が格納されるリポジトリを制御します。netcfgコマンド、NWAM GUI、およびnwamdデーモンはすべて、netcfgdデーモンに要求を送信してリポジトリにアクセスします。これらの機能コンポーネントは、NWAMライブラリ libnwamから要求を行います。

nwamdデーモンは、システムイベントの受信、ネットワークの構成、およびネットワーク構成情報の読み取りを行うポリシーエンジンです。NWAM GUIおよびnetcfgコマンドは、ネットワーク構成を表示および変更する際に使用可能な構成ツールです。これらのコンポーネントは、新しい構成をシステムに適用する必要がある場合に、NWAMサービスを更新する際にも使用されます。
NWAMに関連する承認とプロファイル

現在のNWAM実装では、特定のタスクを実行する際に次の承認が使用されます。

- solaris.network.autoconf.read – netcfgd デーモンで検証されるNWAM構成データの読み取りを有効にします。
- solaris.network.autoconf.write – netcfgd デーモンで検証されるNWAM構成データの書き込みを有効にします。
- solaris.network.autoconf.select – nwamd デーモンで検証される新しい構成データの適用を有効にします。
- solaris.network.autconf.wlan – 既知のWLAN構成データの書き込みを有効にします。

これらの承認は、auth_attrデータベースに登録されています。詳細は、auth_attrのマニュアルページを参照してください。

セキュリティポリシーには、Network Autoconf UserとNetwork Autoconf Adminの2つがあります。Userプロファイルには、read、select、およびwlanの承認があります。Adminプロファイルには、write承認が追加されます。Network Autoconf Userプロファイルは、Console Userプロファイルに割り当てられます。したがって、デフォルトでは、コンソールにログインしているユーザーならだれでも、プロファイルの表示、有効化、および無効化が可能です。Console Userには、solaris.network.autoconf.write承認が割り当てられていないため、このユーザーはNCP、NCU、場所、またはENMを作成または変更できません。ただし、Console UserはWLANを表示、作成、および変更することができます。

NWAMユーザーインターフェースを使用するために必要な承認

NWAMコマンドnetcfgおよびnetadmを使用すると、Console User権を持つユーザーならだれでも、NWAMプロファイルを表示および有効化できます。これらの権限は、/dev/consoleからシステムにログインしているユーザーに自動的に割り当てられます。

netcfgコマンドを使用してNWAMプロファイルを変更するには、solaris.network.autoconf.write承認またはNetwork Autoconf Adminプロファイアルが必要です。

プロファイル名を指定してprofilesコマンドを使用すると、権利プロファイルに関連付けられた特権を特定できます。詳細は、profiles(1)のマニュアルページを参照してください。

たとえば、Console User権利プロファイルに関連付けられた特権を特定するには、次のコマンドを使用します。
NWAM GUI には、特権がない次の 3 つのコンポーネントが含まれています。これら
のコンポーネントには、起動する方法および実行する必要のあるタスクに応じ
て、承認が付与されます。

■ NWAM 固有のパネルの存在
このコンポーネントは、ユーザーが NWAM と対話することを可能にするデスク
トップ上のバネルアプレットです。このパネルは、すべてのユーザーが実行可能
であり、システムの自動構成を監視したり、イベント通知を処理したりする際に
使用されます。このパネルを使用して、WiFi ネットワークの選択や、手動による
場所の切り替えなどの一部の基本ネットワーク構成タスクを実行することもでき
ます。このような種類のタスクを実行するには、Network Autoconf User 権利プロ
ファイルが必要です。パネルは、/dev/console からログインしていて Console
User プロファイルを持っているユーザーの承認で実行されているため、この権利
プロファイルはデフォルトの構成で使用可能です。

■ NWAM GUI
NWAM GUI は、デスクトップから NWAM と対話するための主要な手段で
す。GUI を使用して、ネットワークステータスの表示、NCP と場所プロファイル
の作成と変更、および構成済み ENM の開始と終了を行います。GUI と対話する
には、4 つの solaris.network.autoconf 承認または Network Autoconf Admin プロ
ファイルが必要です。デフォルトでは、Console User プロファイルに、GUI を使
用してネットワークステータスおよびプロファイルを表示するために必要な承認
が含まれています。さらに、GUI を使用してプロファイルを変更するに
は、solaris.network.autoconf.write 承認または Network Autoconf Admin プロ
ファイルが必要です。
追加の承認を取得するには、次の方法のいずれかを実行します。

■ 特定のユーザーに Network Autoconf Admin プロファイルを割り当てます。
特定のユーザーの /etc/user_attr ファイルを編集すれば、直接そのユーザーに適
切な承認または権利プロファイルを割り当てることができます。

■ Console User に Network Autoconf Admin プロファイルを割り当てます。
デフォルトで割り当てられる Network Autoconf User プロファイルの代わりに、こ
のプロファイルを Console User に割り当てることもできます。このプロファイル
を割り当てるには、/etc/security/prof_attr ファイルのエントリを編集します。
NWAM プロファイルの構成(タスク)

この節では、netcfg コマンドを使用して実行可能な NWAM プロファイルの構成タスクについて説明します。これらの構成タスクには、プロファイルの作成、変更、および削除に加えて、NWAM 構成を制御するさまざまな SMF サービスの管理が含まれます。この章では、netcfg コマンドを対話型モードとコマンド行モードの両方で使用する方法について説明します。

この章の内容は次のとおりです。

- 78 ページの「プロファイルを作成する」
- 98 ページの「プロファイルを削除する」
- 100 ページの「プロファイルのプロパティ値を設定および変更する」
- 102 ページの「プロファイル情報についてシステムに問い合わせる」
- 108 ページの「プロファイル構成をエクスポートおよび復元する」
- 112 ページの「ネットワーク構成の管理」

netadm コマンドを使用したプロファイル状態の表示、プロファイルのアクティブ化と非アクティブ化、および既知の無線ネットワークの管理については、第 5 章「NWAM プロファイルの管理(タスク)」を参照してください。

デスクトップから NWAM と対話する方法およびネットワーク構成を管理する方法については、第 6 章「NWAM グラフィカルユーザーアンタフェースについて」を参照してください。

NWAM の紹介については、第 2 章「NWAM の紹介」を参照してください。

netcfg ユーザーアンタフェースモードの説明を含む、NWAM に関する詳細な概要情報については、第 3 章「NWAM 構成と管理(概要)」を参照してください。
プロファイルを作成する

netcfg(1M) のマニュアルページで説明されている netcfg コマンドは、NWAM コマンド行インタフェースの 2つの管理コマンドの 1つです。

netcfg コマンドを使用すると、Console User 特権を持つユーザーであればそれでも、プロファイア構成データを表示したり、既知の WLAN オブジェクトを表示、作成、および変更したりできます。これらの特権は、/dev/console からシステムログインしているユーザーに自動的に割り当てられます。Network Autoconf Admin プロファイルを持つユーザーでは、すべてのタイプの NWAM プロファイルと構成オブジェクトを作成および変更することもできます。詳細は、73 ページの「NWAM セキュリティーの概要」を参照してください。

netcfg コマンドを使用すると、ユーザー定義のプロファイルを選択、作成、変更、および削除できます。このコマンドは、対話型モードまたはコマンド行モードで使用できます。netcfg コマンドでは、プロファイア構成情報のコマンドファイルへのエクスポートもサポートされています。

次のプロファイアおよび構成オブジェクトを作成、変更、および削除できます。

- ネットワーク構成プロファイア (NCP)
- 場所プロファイア
- 外部ネットワーク修飾子 (ENM)
- 既知の無線ローカルエリアネットワーク (WLAN)
- ネットワーク構成ユニット (NCU)

プロファイルをコマンド行モードで作成する

プロファイルを作成するためのコマンドは次のとおりです。

netcfg create [ -t template ] object-type [ class ] object-name

create 指定されたタイプおよび名前のインメモリープロファイア (または構成オブジェクト) を作成します。

-t template 新しいプロファイアが template と同一であることを指定します。ここで、template は同じタイプの既存のプロファイアの名前を表します。-t オプションを使用しない場合、新しいプロファイアはデフォルト値で作成されます。

object-type 作成されるプロファイアのタイプを指定します。

object-type オプションには、次の値のいずれかを指定できます。

- ncp
- ncu
プロファイルを作成する

- loc
- enm
- wlan

netcfg select コマンドを使用して特定のオブジェクトを選択する前に、ncu を除く、object-type オプションで指定されたすべてのプロファイルをグローバルスコープで作成する必要があります。

class object-type で指定されたプロファイルのクラスを指定します。このパラメータは、ncu オブジェクト型でのみ使用され、指定可能な値は phys または ip の 2 つです。

object-name ユーザー定義のプロファイルの名前を指定します。NCU の場合、object-name は対応するリンクまたはインタフェースの名前です。その他のすべてのプロファイルタイプの場合、object-name は任意のユーザー定義の名前です。

たとえば、User という名前の NCP を作成するには、次のコマンドを入力します。

$ netcfg create ncp User

ここで、ncp は object-type、User は object-name を表します。

注-NCP を作成する場合は、class オプションは必要ありません。

ここで示すように、必要に応じて Automatic NCP のコピーをテンプレートとして使用すると、そのプロファイルに変更を加えることができます。

$ netcfg create -t Automatic ncp

名前が office の場所プロファイルを作成するには、次のコマンドを入力します。

$ netcfg create loc office

プロファイルを対話形式で作成する

netcfg コマンドを対話型モードで使用すると、次のタスクを実行できます。

- プロファイルを作成します。
- プロファイルを選択して変更します。
- プロファイルに関する必須情報がすべて設定され、有効であることを確認します。
- 新しいプロファイルの変更を確定します。
プロファイルを作成する

- 永続ストレージで変更を確定しないで、現在のプロファイル構成を取り消します。
- プロファイルに行った変更を元に戻します。

**NCP を作成する**

プロファイルを対話型モードで作成すると、コマンドプロンプトが次のスコープのいずれかになります。
- NCP が作成された場合、NCP スコープ
- 場所プロファイル、ENM、または WLAN オブジェクトが作成された場合、プロファイルスコープ

NCP または NCU を作成すると、フォーカスがオブジェクトのスコープに移動し、指定されたプロファイルのデフォルトプロパティが調査されます。

NCP を対話形式で作成するには、netcfg 対話型セッションを開始します。次に、create サブコマンドを使用して、次のように User という新規 NCP を作成します。

```bash
$ netcfg
netcfg> create ncp User
netcfg:ncp:User>
```

**NCP に NCU を作成する**

NCP は基本的に、一連の NCU で構成されるコンテナです。すべての NCP には、リンクとインタフェースの両方の NCU が含まれています。リンク NCU には、リンク構成とリンク選択ポリシーの両方を指定します。インタフェース NCU には、インタフェース構成ポリシーを指定します。IP 接続が必要な場合は、リンクとインターフェースの両方の NCU が必要です。netcfg コマンドまたは GUI を使用して、NCU を明示的に追加または削除する必要があります。

注 - 現在システムにインストールされているリンクとは関連性のない NCU を追加することができます。さらに、現在システムにインストールされているリンクにマップしている NCU を削除することもできます。

NCU を作成するには、netcfg コマンドを対話型モードまたはコマンド行モードで使用します。NCU の作成には複数の操作が伴うため、NCU およびすべてのプロパティーを作成する単一行のコマンドを構築しようとするよりも、対話型モードを使用した方が簡単かつ効率的に NCU を作成できます。NCU は、NCP の初期作成時ま

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月 80
たは作成後に作成できます。NCU を作成または変更するプロセスには、一般的な NCU プロパティーの設定に加えて、各 NCU タイプに特に適用されるプロパティーの設定も含まれます。

NCP に NCU を作成するプロセス中に表示されるプロパティーは、特定の NCP の作成中に行われた選択に基づいて最適化されます。

NCU を対話形式で作成すると、netcfg によって関連する各プロパティーが調査され、デフォルト値(デフォルトが存在する場合)と指定可能な値の両方が表示されます。値を指定しないで Return キーを押すと、デフォルト値が適用されます(デフォルト値がない場合、プロパティーは空白のままです)。また、代替の値を指定することもできます。NCP に NCU を作成するプロセス中に表示されるプロパティーは、すでに行われた選択に応じています。たとえば、インタフェース NCU の ipv4-addrsrc プロパティーで dhcp を選択すると、ipv4-addr プロパティーの値を指定するよう求めるプロンプトは表示されません。

次の表では、NCU の作成または変更時に指定する可能性のあるすべての NCU プロパティーについて説明します。一部のプロパティーは、両方の NCU タイプに適用されます。その他のプロパティーは、リンク NCU とインタフェース NCU のどちらかに適用されます。すべての NCU プロパティーの詳細(これらのプロパティーの指定時に適用される可能性のある規則や条件を含む)は、netcfg(1M) のマニュアルページを参照してください。

**表4-1 NCU を作成または変更する NCU プロパティー**

<table>
<thead>
<tr>
<th>プロパティー</th>
<th>説明</th>
<th>取り得る値</th>
<th>NCU タイプ</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>NCU タイプを、link または interface のどちらかに指定します。</td>
<td>link または interface</td>
<td>リンクとインタフェース</td>
</tr>
<tr>
<td>class</td>
<td>NCU クラスを指定します。</td>
<td>phys(リンク NCU) または ip(インタフェース NCU)</td>
<td>リンクとインタフェース</td>
</tr>
<tr>
<td>parent</td>
<td>この NCU が属する NCP を指定します。</td>
<td>parent-NCP</td>
<td>リンクとインタフェース</td>
</tr>
<tr>
<td>enabled</td>
<td>NCU を有効にするか、無効にするかを指定します。</td>
<td>true または false</td>
<td>リンクとインタフェース</td>
</tr>
</tbody>
</table>

第4章・NWAM プロファイルの構成(タスク) 81
プロファイルを作成する

表4-1 NCUを作成または変更する NCU プロパティ（続き）

<table>
<thead>
<tr>
<th>プロパティ</th>
<th>説明</th>
<th>取り得る値</th>
<th>NCUタイプ</th>
</tr>
</thead>
<tbody>
<tr>
<td>activation-mode</td>
<td>NCUの自動アクティブ化のトリガータイプを指定します。</td>
<td>manual または prioritized</td>
<td>リンク</td>
</tr>
<tr>
<td></td>
<td>デフォルト値はmanualです。</td>
<td></td>
<td></td>
</tr>
<tr>
<td>priority-group</td>
<td>グループの優先度番号を指定します。</td>
<td>0 (有線リンク)または1 (無線リンク)</td>
<td>リンク</td>
</tr>
<tr>
<td></td>
<td>ユーザー定義のNCPでは、さまざまなポリシーを指定できます（たとえば、無線リンク1は優先度1、有線リンク1は優先度2、有線リンク2は優先度3）。</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>注-番号が小さいほど、高い優先度を示します。</td>
<td></td>
<td></td>
</tr>
<tr>
<td>priority-mode</td>
<td>activation-modeプロパティをprioritizedに設定した場合、優先グループのアクティブ化動作を決定する際使用されるモードを指定します。</td>
<td>exclusive, shared、または all</td>
<td>リンク</td>
</tr>
<tr>
<td></td>
<td>これらの値を指定する際、有線リンク1は優先度1、有線リンク2は優先度2、有線リンク3は優先度3。</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link-mac-addr</td>
<td>このリンクに割り当てられるMACアドレスを指定します。デフォルトでは、工場出荷時に割り当てられたMACアドレス、またはその他のデフォルトMACアドレスがNWAMで使用されます。ここで別の値を設定して、これらの選択を上書きすることができま</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>48ビットのMACアドレスを含む文字列</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link-autopush</td>
<td>開いたときに自動的にリンクでプッシュされるモジュールを指定します。</td>
<td>文字列のリスト(リンクでプッシュされるモジュール)</td>
<td>リンク</td>
</tr>
<tr>
<td></td>
<td>autopush(1M)を参照してください。</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
表4-1 NCUを作成または変更するNCUプロパティー（続き）

<table>
<thead>
<tr>
<th>プロパティー</th>
<th>説明</th>
<th>取り得る値</th>
<th>NCUタイプ</th>
</tr>
</thead>
<tbody>
<tr>
<td>link-mtu</td>
<td>物理リンクにはデフォルトのMTUが自動的に設定されます。これを別に設定すると、値を上書きすることが可能。</td>
<td>リンクのMTUサイズ</td>
<td>リンク</td>
</tr>
<tr>
<td>ip-version</td>
<td>使用するIPのバージョンを指定します。複数の値を割り当てることができます。</td>
<td>ipv4とipv6</td>
<td>インタフェース</td>
</tr>
<tr>
<td>ipv4-addrsrc</td>
<td>このNCUに割り当てられたIPv4アドレスの発信元を指定します。複数の値を割り当てることができます。</td>
<td>dhcpとstatic</td>
<td>インタフェース</td>
</tr>
<tr>
<td>ipv6-addrsrc</td>
<td>このNCUに割り当てられたIPv6アドレスの発信元を指定します。複数の値を割り当てることができます。</td>
<td>dhcp、autoconf、またはstatic</td>
<td>インタフェース</td>
</tr>
<tr>
<td>ipv4-addr</td>
<td>このNCUに割り当てられる1つまたは複数のIPv4アドレスを指定します。</td>
<td>割り当てられる1つまたは複数のIPv4アドレス</td>
<td>インタフェース</td>
</tr>
<tr>
<td>ipv6-addr</td>
<td>このNCUに割り当てられる1つまたは複数のIPv6アドレスを指定します。</td>
<td>割り当てられる1つまたは複数のIPv6アドレス</td>
<td>インタフェース</td>
</tr>
<tr>
<td>ipv4-default-route</td>
<td>IPv4アドレスのデフォルトルートを指定します。</td>
<td>IPv4アドレス</td>
<td>インタフェース</td>
</tr>
<tr>
<td>ipv6-default-route</td>
<td>IPv6アドレスのデフォルトルートを指定します。</td>
<td>IPv6アドレス</td>
<td>インタフェース</td>
</tr>
</tbody>
</table>

▼ NCPを対話形式で作成する方法

次の手順では、NCPを対話型モードで作成する方法について説明します。
プロファイルを作成する

ヒント - プロファイルの初期作成中にNWAMで実行される調査プロセスでは、事前に行われた選択に基づき意味があるプロパティーに関するプロンプトのみが表示されます。また、この手順で説明するverifyサブコマンドは、構成を検証します。必要な値が見つからない場合は、通知されます。プロファイルの作成または変更時にverifyサブコマンドを使用すると明示的に、commitサブコマンドを使用すると暗黙的に、変更を保存できます。

1 netcfg対話型セッションを開始します。

   $ netcfg
   netcfg>

2 NCPを作成します。

   netcfg> create ncp User
   netcfg:ncp:User>

ここで、ncpはプロファイルタイプ、Userはプロファイル名を表します。

NCPを作成すると、自動的にNCPスコープに移動します。場所、ENM、またはWLANオブジェクトを作成した場合は、コマンドプロンプトでプロファイルスコープに移動します。

3 NCPにリンクNCUおよびインタフェースNCUを作成します。

a. リンクNCUを作成するには、次のコマンドを入力します。

   netcfg:ncp:User> create ncu phys net0
   Created ncu 'net0', Walking properties ...

   ここで、ncuはオブジェクト型、physはクラス、net0(例示の目的のみ)はオブジェクト名を表します。

   NCUを作成すると、オブジェクトのスコープに移動し、オブジェクトのデフォルトプロパティーが調査されます。

b. インタフェースNCUを作成するには、次のコマンドを入力します。

   netcfg:ncp:User> create ncu ip net0
   Created ncu 'net0', walking properties ...

   ここで、ncuはオブジェクト型、ipはクラス、net0(例示の目的のみ)はオブジェクト名を表します。

   NCUを作成すると、オブジェクトのスコープに移動し、オブジェクトのデフォルトプロパティーが調査されます。

   NCUの作成中は、2つのNCUタイプを区別するためにclassオプションが使用されます。このオプションは、異なるNCUタイプで同じ名前を共有する状況で特に役立ちます。classオプションを省略すると、同じ名前を共有するNCUを区別することがかなり難しくなります。
4 作成したNCUに適切なプロバティーを追加します。

注 - NCPに必要なNCUがすべて作成されるまで、手順3と4を繰り返します。

5 NCUの作成中、または指定されたNCUのプロバティー値の設定時に、verifyサブコマンドを使用して、行われた変更が正しいことを確認します。

netcfg:ncp:User:ncu:net0> verify
All properties verified

6 NCUに設定したプロバティーを確定します。

netcfg:ncp:User:ncu:net0> commit
committed changes.

または、endサブコマンドを使用して、暗黙の確定処理を実行することもできます。
これにより、対話型セッションは1つ上のレベルである次のスコープに移動します。この場合、NCUの作成とNCUの追加が完了したら、対話型セッションをNCPスコープから直接終了できます。

注 -
- 対話型モードでは、確定されるまで変更は永続ストレージに保存されません。
  commitサブコマンドを使用すると、プロファイル全体が確定されます。永続ストレージの整合性を維持するために、確定操作には検証手順も含まれます。
  検証に失敗すると、確定処理にも失敗します。暗黙の確定処理に失敗すると、現在の変更を確定せずに対話型セッションを終了するオプションが与えられます。また、現在のスコープに残って、プロファイルの変更を続行することもできます。
- 行った変更を取り消すには、cancelまたはrevertサブコマンドを使用します。
  cancelサブコマンドは、現在の変更を永続ストレージに確定せずに現在のプロファイル構成を終了します。これにより、対話型セッションは1レベル高い次のスコープに移動します。
  revertサブコマンドは、行った変更を元に戻し、以前の構成を再度読み込みます。
  revertサブコマンドを使用すると、対話型セッションは同じスコープに残ります。

7 NCPの構成を表示するには、listサブコマンドを使用します。

8 NCPの構成が完了したら、対話型セッションを終了します。

netcfg:ncp:User> exit

exitサブコマンドを使用して、netcfg対話型セッションを終了すると、常に現在のプロファイルが検証され、確定されます。検証と確定のどちらかの操作に失敗すると、適切なエラーメッセージが発行され、現在の変更を確定せずに終了する機会が与えられます。
または、現在のスコープに残って、プロファイルの変更を続行することもできます。
例4-1 NCPを対話形式で作成する

次の例では、1つのNCPと2つのNCU(1つのリンクと1つのインタフェース)が作成されます。

```bash
$ netcfg
netcfg> create ncp User
netcfg:ncp:User> create ncu phys net0
Created ncu 'net0', Walking properties ...
activation-mode (manual) [manual|prioritized]>
link-mac-addr>
link-autopush>
link-mtu>
netcfg:ncp:User:ncu:net0> end
Committed changes
netcfg:ncp:User> create ncu ip net0
Created ncu 'net0', Walking properties ...
ip-version (ipv4,ipv6) [ipv4|ipv6]>
ipv4
ipv4-addrsrc (dhcp) [dhcp|static]>
ipv4-default-route>
netcfg:ncp:User:ncu:net0> verify
All properties verified
netcfg:ncp:User:ncu:net0> end
Committed changes
netcfg:ncp:User> list
NCUs:
  phys net0
  ip net0
netcfg:ncp:User> list ncu phys net0
ncu:net0
  type link
class phys
parent "User"
activation-mode manual
enabled true
netcfg:ncp:User> list ncu ip net0
ncu:net0
  type interface
class ip
parent "User"
enabled true
ip-version ipv4
ipv4-addrsrc dhcp
ipv6-addrsrc dhcp,autoconf
netcfg:ncp:User> exit
```

注 - netcfg対話型セッションを終了せずにスコープを終了するには、endコマンドを入力します。

```bash
netcfg:ncp:User> end
netcfg>
```
例4-2 既存のNCPにNCUを作成する

既存のNCPにNCUを作成したり、既存のプロファイルのプロパティを変更するには、selectサブコマンドとともにnetcfgコマンドを使用します。

次の例では、既存のNCPにIP NCUが作成されます。既存のプロファイルを対話型モードで変更するプロセスは、プロファイルの作成と似ています。例4-1との違いは、この例ではNCPがすでに存在するため、createサブコマンドではなく、selectサブコマンドが使用される点です。

```
$ netcfg
netcfg> select ncp User
netcfg:ncp:User> list
NCUs:
    phys net0
netcfg:ncp:User> create ncu ip net0
Created ncu 'net0'. Walking properties ...
ip-version (ipv4,ipv6) [ipv4|ipv6]> ipv4
ipv4-addrsrc (dhcp) [dhcp|static]> dhcp
ipv4-default-route> netcfg:ncp:User:ncu:net0> end
Committed changes
netcfg:ncp:User> list
NCUs:
    phys net0
    ip net0
netcfg:ncp:User> list ncu phys net0
ncu:net0
    type link
    class phys
    parent "User"
    activation-mode manual
    enabled true
netcfg:ncp:User> list ncu ip net0
NCU:net0
    type interface
    class ip
    parent "User"
    enabled true
    ip-version ipv4
    ipv4-addrsrc dhcp
    ipv6-addrsrc dhcp,autoconf
netcfg:ncp:User> exit
$
場所プロファイルを作成する

場所プロファイルには、基本リンクおよびIP接続に直接関連しないネットワーク構成の設定を定義するプロパティが含まれています。一部の例には、必要に応じて、同時に適用されるネームサーバおよびIPフィルタの設定が含まれています。常に、システムで1つの場所プロファイルと1つのNCPをアクティブにする必要があります。システム定義の場所とユーザー定義の場所があります。システムの場所は、場所を指定しなかった場合や、手動でアクティブ化された場所が有効でない場合などの特定の条件で、NWAMによって選択されるデフォルトであり、条件付きでアクティブ化された場所の条件はどれも満たしていません。システム定義の場所の起動モードは、systemです。ユーザー定義の場所は、ネットワーク接続によって取得されるIPアドレスなどのネットワーク状況に応じて、手動または条件付きでアクティブ化されるように構成された場所です。

場所プロファイルの手動アクティブ化(有効化)については、118ページの「プロファイルをアクティブまたは非アクティブにする」を参照してください。

場所を作成するには、netcfg コマンドを対話型モードまたはコマンド行モードで使用します。場所プロファイルを作成する際は、その場所に特定の構成パラメータを定義する値を指定することによって、場所のプロパティを設定する必要があります。場所のプロパティは、特定クラスの構成設定を示すグループ別に分類されます。

場所のプロパティは、NWAMによってリポジトリにも格納されます。特定の場所プロファイルがアクティブになると、NWAMはその場所に設定されたプロパティに基づいて、ネットワークを自動的に構成します。場所の作成または変更には、プロファイルの構成方法を定義するさまざまなプロパティの設定が伴います。これにより、NWAMによるネットワークの自動構成の方法が決まります。構成プロセス中で表示されるプロパティは、事前に行われた選択に基づいて、最適化されたプロパティとなります。

次の表では、指定可能なすべての場所のプロパティについて説明します。場所のプロパティはグループ別に分類されていることに注意してください。すべての場所のプロパティの詳細(プロパティの指定時に適用される可能性のある規則、条件、依存など)は、netcfg(1M)のマニュアルページを参照してください。

表4-2 場所のプロパティと説明

<table>
<thead>
<tr>
<th>プロパティーグループと説明</th>
<th>プロパティ値と説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>選択条件</td>
<td>activation-mode</td>
</tr>
<tr>
<td>場所をアクティブまたは非アクティブにする方法とタイミングを指定します。</td>
<td>manual、conditional-any、およびconditional-allです。</td>
</tr>
</tbody>
</table>

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化 - 2011年12月
プロファイルを作成する

<table>
<thead>
<tr>
<th>プロパティーグループと説明</th>
<th>プロパティー値と説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>システムドメイン</td>
<td>system-domainプロパティーは、default-domainプロパティーで構成されます。このプロパティーは、リモートプロシージャコール(RPC)の変換で使用されるシステム全体のドメインを指定します。</td>
</tr>
<tr>
<td>NISネームサービスで直接使用されるホストのドメイン名を指定します。</td>
<td>ネームサービスの情報を使用するネームサービスおよびネームサービススイッチの構成を指定します。</td>
</tr>
</tbody>
</table>

指定されたネームサービスのプロパティーを一覧表示します。
- domain-name
- nameservices
- nameservices-config-file
- dns-nameservice-configsrc
- dns-nameservice-domain
- dns-nameservice-servers
- dns-nameservice-search
- dns-nameservice-sortlist
- dns-nameservice-options
- nis-nameservice-configsrc
- nis-nameservice-servers
- ldap-nameservice-configsrc
- ldap-nameservice-servers

これらのプロパティーの詳細は、netcfg(IM)のマニュアルページの「場所のプロパティー」節を参照してください。

NFSv4 ドメイン

NFSv4 ドメインを指定します。

システムのnfsmapid_domainプロパティーで使用される値。nfsmapidのマニュアルページで説明するように、場所がアクティブな場合に、この値を使用してnfsmapid_domainSMFプロパティーを設定します。このプロパティーを設定しないと、場所がアクティブな場合、システムのnfsmapid_propertyがクリアされます。詳細は、nfsmapid(IM)のマニュアルページを参照してください。

IPフィルタの構成

IPフィルタの構成で使用されるパラメータを指定します。これらのプロパティーには、IPフィルタとNATの規則が含まれる適切なipfおよびipnatファイルへのパスが指定されます。

- ipfilter-config-file
- ipfilter-v6-config-file
- ipnat-config-file
- ippool-config-file

構成ファイルが指定されている場合は、識別されたファイルに含まれる規則が適切なipfilterサブシステムに適用されます。
プロファイルを作成する

<table>
<thead>
<tr>
<th>IPsecの構成ファイル</th>
<th>IPsecの構成で使用されるファイルを指定します。</th>
</tr>
</thead>
<tbody>
<tr>
<td>□ ike-config-file</td>
<td>□ ipsecpolicy-config-file</td>
</tr>
</tbody>
</table>

### 場所プロファイリを対話形式で作成する方法

次の手順では、場所プロファイリを作成する方法について説明します。

ヒント - プロファイリの初期作成中にNWAMで実行される調査プロセスでは、事前に入力された値に基づき、意味があるプロファイリのブロンドトのみが表示されます。また、verifyサブコマンドでは、構成が正しいかどうかがチェックされます。必要な値が見つからない場合は、通知されます。プロファイリ構成の作成または変更時にverifyサブコマンドを使用すると明示的に、commitサブコマンドを使用すると暗黙的に、変更を保存できることに注意してください。

1. netcfg対話型セッションを開始します。
   ```
s netcfg
netcfg>
```
2. 場所を作成または選択します。
   ```
netcfg> create loc office
netcfg:loc:office>
```
   この例では、officeという場所が作成されます。
   場所を作成すると、この場所のプロファイルスコープに自動的に移動します。
3. 場所に適切なプロバティーを設定します。
4. プロファイル構成を表示します。
   たとえば、次の出力では場所officeのプロバティーが表示されます。
   ```
netcfg:loc:office> list
LOC:office
    activation-mode                      conditional-any
    conditions                           "ncu ip:wpi0 is active"
    enabled                              false
    nameservices                         dns
    nameservices-config-file             "/etc/nsswitch.dns"
    dns-nameservice-configsrc           dhcp
    ipfilter-config-file                 "/export/home/test/wifi.ipf.conf"
```
5 プロファイル構成が正しいかどうかを検証します。
次を例では、場所 "office" の構成が検証されます。

```
netcfg:loc:office> verify
All properties verified
```

6 検証が完了したら、場所プロファイルを永続ストレージで確定します。

```
netcfg:loc:office> commit
Committed changes
```
また、end サブコマンドを使用してセッションを終了して、プロファイル構成を保存することもできます。

```
netcfg:loc:office> end
Committed changes
```

注 -

- 対話型モードでは、変更は確定されるまで、永続ストレージに保存されません。commit サブコマンドを使用すると、プロファイル全体が確定されます。永続ストレージの整合性を維持するために、確定操作には検証手順も含まれます。検証に失敗すると、確定処理にも失敗します。暗黙の確定処理に失敗すると、現在の変更を確定せずに対話型セッションを終了するオプションが与えられます。または、現在のスコープに残って、プロファイルの変更を継続することもできます。
- 行った変更を取り消すには、cancel サブコマンドを使用します。
cancel サブコマンドは、現在の変更を永続ストレージに確定せずに現在のプロファイル構成を終了します。これにより、対話型セッションは1つ上のレベルである次のスコープに移動します。

7 対話型セッションを終了します。

```
netcfg> exit
Nothing to commit
```

例4-3 場所プロファイルを対話形式で作成する

次の例では、"office" という場所が作成されます。

```
$ netcfg
netcfg> create loc office
Created loc 'office'. Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-any
conditions> ncu ip:wpid is active
nameservices (dns) [dns|files|nis|ldap]>
nameservices-config-file ("/etc/nsswitch.dns")>
dns-nameservice-configsrc (dhcp) [manual|dhcp]>
fsv4-domain>
```

第4章・NWAMプロファイルの構成(タスク)
プロファイルを作成する

```bash
ipfilter-config-file> /export/home/test/wifi.ipf.conf
ipfilter-v6-config-file>
ipnat-config-file>
ippool-config-file>
ike-config-file>
ipsec-policy-config-file>
netcfg:loc:office> list
LOC:office
  activation-mode               conditional-any
  conditions                   "ncu ip:wpi0 is active"
  enabled                      false
  nameservices                 dns
  nameservices-config-file     "/etc/nsswitch.dns"
  dns-nameservice-configsrc   dhcp
  ipfilter-config-file         "/export/home/test/wifi.ipf.conf"
netcfg:loc:office> verify
All properties verified
netcfg:loc:office> commit
Committed changes
netcfg> list
NCPs:
  User
  Automatic
Locations:
  Automatic
  NoNet
  test-loc
WLANs:
  sunwifi
  ibahn
  gogoinflight
  admiralsclub
  hhonors
  sjcfreewifi
netcfg> exit
Nothing to commit
```

この例では、次のプロパティが場所officeに指定されました。

- **activation-mode** プロパティが**conditional-any**に設定されたため、アクティブ化の条件を指定できるコマンドプロンプトが表示されました。
- アクティブ化の条件は、*ncu ip:wpi0 is active* と指定されました。

注 - 前の手順で**conditional-any** プロパティが指定されたため、**conditions** プロパティが要求されました。たとえば、**manual** プロパティが指定された場合は、**conditions** プロパティは要求されません。

- **Return** キーを押すことにより、次のデフォルト値が受け入れられました。
  - nameservices
  - nameservices-config-file
  - dns-nameservice-configsrc

---

Oracle Solaris管理:ネットワークインタフェースとネットワーク仮想化・2011年12月
ENM プロファイルを作成する

ENM は、VPN アプリケーションなど NWAM 外部のアプリケーションの構成に関連します。これらのアプリケーションでは、ネットワーク構成の作成および変更が可能です。ENM は、アクティブまたは非アクティブにするとネットワーク構成を直接変更するサービスやアプリケーションとして定義することもできます。指定された条件で ENM がアクティブまたは非アクティブになるように、NWAM を構成できます。システム上で常に各プロファイルタイプの 1 つのみをアクティブにできる NCP または場所プロファイルとは異なり、同時に複数の ENM をシステム上でアクティブにすることができる可能性があります。システム上で常にアクティブな ENM は、システム上で同時にアクティブな NCP または場所プロファイルに必ずしも依存しません。

注: NWAM では、ENM を作成する可能性のあるアプリケーションを自動的には認識しません。netcfg コマンドを使用して ENM を作成する前に、まずこれらのアプリケーションをシステムにインストールして構成する必要があります。

ENM を作成するには、次のコマンドを入力します。

```bash
$ netcfg
netcfg> create enm my_enm
Created enm ‘my_enm’. Walking properties ...
```
ここで、enmはENMプロファイル、my_enmはオブジェクト名を表します。

ENMを作成するプロセスでは、新たに作成されたENMのプロファイルスコープに
移動し、新たに作成されたENMでプロパティの調査が自動的に開始されます。こ
こから、ENMをアクティブにするタイミングと方法を指示するENMのプロパ
ティー、およびENMの開始メソッドや終了メソッドなどのその他の条件を設定でき
ます。

ENMプロパティを指定する詳細な手順については、netcfg(1M)のマニュアル
ページを参照してください。

次の表では、ENMの作成または変更時に指定されるプロパティについて説明しま
す。

<table>
<thead>
<tr>
<th>プロパティ名</th>
<th>説明</th>
<th>取り得る値</th>
</tr>
</thead>
<tbody>
<tr>
<td>activation-mode</td>
<td>ENMのアクティブ化を決定する際に使用されるモード</td>
<td>conditional-any、</td>
</tr>
<tr>
<td></td>
<td></td>
<td>conditional-all、 manual</td>
</tr>
</tbody>
</table>
| conditions    | activation-modeがconditional-anyまたはconditional-allの場合に、ENMをアクティブにする必要があるかどうかを判定するためのテストを指定します。 | このプロパティが使用されている場合、netcfg(1M)のマニュアルページの「条件式」節で指定された形式の文字列。
| start         | (省略可能)アクティブ化時に実行されるスクリプトへの絶対パス             | このプロパティが使用されている場合、スクリプトへのパス。
| stop          | (省略可能)非アクティブ化時に実行されるスクリプトへの絶対パス           | このプロパティが使用されている場合、スクリプトへのパス。
| fmri          | (省略可能)ENMのアクティブ化時に有効になるFMRI(障害管理リソース識別子) | スクリプトへのパス。            |
|               | 注-FMRIと開始スクリプトのどちらかを指定する必要がありれます。FMRIが指定されている場合は、startとstopプロパティはどちらも無視されます。 |

例4-4 ENMプロファイルを対話形式で作成する
次の例では、test-enmという名前のENMが対話型モードで作成されます。

```bash
$ netcfg
netcfg> create enm test-enm
Created enm 'testenm'. Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-all]>
fmri> svc:/application/test-app:default
start>
```
例 4-4 ENM プロファイルを対話形式で作成する（続き）

stop>
netcfg:enm:test-enm> list
ENM:test-enm
   activation-mode  manual
   enabled          false
   fmri            "svc:/application/test-enm:default"
netcfg:enm:test-enm> verify
All properties verified
netcfg:enm:test-enm> end
Committed changes
netcfg> list
NCPs:
   User
   Automatic
Locations:
   Automatic
   NoNet
   test-loc
ENMs:
   test-enm
WLANs:
   sunwifi
   ibahn
   gogoinflight
   admiralsclub
   hhonors
   sjcfreewifi
netcfg> end
$

この例では、次のプロパティ値を使用して test-enm という名前の ENM が作成されました。

- Return キーを押して、activation-mode プロパティのデフォルト値（manual）が受け入れられました。
- アプリケーションのアクティブ化および非アクティブ化で使用されるメソッドとして、SMF FMRI プロパティ svc:/application/test-enm:default が指定されました。
- FMRI が指定されたため、start および stop メソッドプロパティはバイパスされました。
- list サブコマンドを使用して、ENM のプロパティを表示しました。
- verify サブコマンドを使用して、プロファイル構成が正しいことを確認しました。
- end サブコマンドを使用して、暗黙的に構成を保存しました。
- end サブコマンドを再度使用して、対話型セッションを終了しました。
WLANを作成する

NWAMでは、システム全体の既知のWLANのリストが保持されます。WLANは、システムに接続された無線ネットワークの履歴および構成情報が含まれる構成オブジェクトです。このリストは、NWAMが使用可能な無線ネットワークへの接続を試みる順序を決定する際に使用されます。「Known WLAN」リストに存在する無線ネットワークが使用可能な場合、NWAMは自動的にそのネットワークに接続します。既知のネットワークが2つ以上使用可能な場合、NWAMは優先度がもっとも高い（番号がもっとも小さい）無線ネットワークに接続します。NWAMが接続する新しい無線ネットワークは「既知のWLAN」リストの一番上に追加され、その新しい無線ネットワークの優先度がもっとも高くなります。

WLANオブジェクトを作成するには、次のコマンドを入力します。

```
$ netcfg
netcfg> create wlan mywifi
Created wlan 'mywifi'. Walking properties ...
```

ここで、wlanはWLANオブジェクト、mywifiはオブジェクト名を表します。

WLANオブジェクトを作成するプロセスでは、新たに作成されたWLANのプロファイルスコープに移動し、新たに作成されたWLANでプロバティーの調査が自動的に開始されます。ここから、WLANの構成を定義するためのプロバティーを設定できます。

次の表では、WLANの作成または変更時に指定されるプロバティーについて説明します。

<table>
<thead>
<tr>
<th>既知のWLANプロバティー</th>
<th>プロバティーのデータ型</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>ESSID (無線ネットワーク名)</td>
</tr>
<tr>
<td>bssids</td>
<td>指定されたWLANへの接続時にシステムが接続したWLANのベースステーションID</td>
</tr>
<tr>
<td>priority</td>
<td>WLAN接続の優先順位 (小さい値が優先)</td>
</tr>
<tr>
<td>keyslot</td>
<td>WEPキーが含まれるスロット番号 (1-4)</td>
</tr>
<tr>
<td>keyname</td>
<td>dladm create-secobjコマンドを使用して作成されるWLANキーの名前</td>
</tr>
<tr>
<td>security-mode</td>
<td>使用される暗号化鍵のタイプ。タイプはnone、wep、またはwpaです。</td>
</tr>
</tbody>
</table>

例4-5 WLANを作成する
次の例では、mywifiという名前のWLANオブジェクトが作成されます。
例 4-5  WLANを作成する  （続き）

この例では、WLAN を追加する前に、WLAN mywifi の keyname プロパティで指定されたキーが含まれる、mywifi-key という名前のセキュリティー保護されたオブジェクトが作成されていることを前提としています。

その他の WLAN が追加または削除されると、優先度番号が変更される可能性があります。2 つの WLAN に同一の優先度番号を割り当てることはできないことに注意してください。優先される WLAN に関しては、番号が小さいほど優先度が高くなります。この例では、その他の既知の WLAN よりも優先度が低くなるように、WLAN に優先度番号 100 が割り当てられます。

手順の最後で list サブコマンドを使用すると、新しい WLAN がリストの一番下に追加されます。これは、既存の既知の全 WLAN の中で優先度がもっとも低いことを示します。WLAN に優先度番号としてデフォルトのゼロ (0) が割り当てられた場合、リストの一番上に表示されます。これは、優先度がもっとも高いことを示します。その結果、その他の既知のすべての WLAN の優先度が下げられ、リストでは新たに追加された WLAN のあとに表示されます。

```
$ netcfg
netcfg> create wlan mywifi
Created wlan 'mywifi'. Walking properties ... priority (0)> 100
bssids>
keyname> mywifi-key
keyslot>
security-mode [none|wep|wpa] wpa
netcfg:wlan:mywifi> list
WLAN:mywifi
  priority 100
  keyname "mywifi-key"
  security-mode wpa
netcfg:wlan:mywifi> verify
All properties verified
netcfg:wlan:mywifi> end
Committed changes
netcfg> list
NCPs:
  User
  Automatic
Locations:
  Automatic
  NoNet
  test-loc
ENMs:
  test-enm
WLANs:
  sunwifi
  ibahn
  gogo_inflight
  admiralsclub
  hhonors
  sjcfreewifi
```
プロファイルを削除する

例4-5 WLANを作成する (続き)

```bash
mywifi
netcfg> exit
Nothing to commit
$```

プロファイルを削除する

```
netcfg destroy -a コマンドを使用すると、すべてのユーザー定義のプロファイルまたは指定されたユーザー定義のプロファイルをメモリおよび永続ストレージから削除できます。
```

注 - システム定義のプロファイル (Automatic NCP、および「NoNet」と「Automatic」の場合プロファイルを含む) は削除できません。

```
destroy コマンドの構文は、次のとおりです。

```bash
netcfg destroy object-type [ class ] object-name
```

また、次のコマンドを使用すると、システム内のユーザー定義のプロファイルをすべて削除することもできます。

```
netcfg destroy -a
```

例4-6 netcfg コマンド行モードを使用して、ユーザー定義のプロファイルをすべて削除するシステム上のユーザー定義のプロファイルをすべて削除するには、次のコマンドを入力します。

```
$ netcfg destroy -a
```

システムでは、常に1つ以上のプロファイルをアクティブにする必要があるため、ユーザー定義のプロファイルの削除時に使用中のエラーを回避するために、destroy -a コマンドを使用する前に、Automatic NCP が有効になっていることを確認します。

例4-7 netcfg コマンド行モードを使用して、特定のユーザー定義のプロファイルを削除するシステム上の特定のユーザー定義のプロファイル (User という名前の NCP など) を削除するには、次のコマンドを入力します。

```
$ netcfg destroy ncp User
```

destroy コマンドを使用すると、既存の NCP から NCU を削除することもできます。次の例では、net1 という名前のインタフェース NCU がユーザー定義の NCP から削除されます。

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化  2011年12月
例4-7 netcfgコマンド行モードを使用して、特定のユーザ定義のプロファイルを削除する（続き）

```bash
$ netcfg "select ncp User; destroy ncu ip net1"
プロファイルが削除されたことを確認するには、ここで示すように、listサブコマンドを使用します。

```bash
$ netcfg
netcfg> select ncp User
netcfg:ncp:User> list
NCUs:
   phys net1
netcfg> exit
Nothing to commit
$
```

例4-8 プロファイルを対話形式で削除する
次の例では、net2という名前のIP NCUが削除されます。

```bash
$ netcfg list
NCPs:  
   Automatic
   User
Locations:  
   Automatic
   NoNet
   test
   foo
$ netcfg
netcfg> select ncp User
netcfg:ncp:User> list
NCUs:
   phys net2
   ip net2
netcfg:ncp:User> destroy ncu ip net2
Destroyed ncu ‘net2’
netcfg:ncp:User> list
NCUs:
   phys net2
netcfg:ncp:User> end
netcfg> exit
Nothing to commit
$
```
プロファイルのプロバティー値を設定および変更する

新規および既存のユーザー定義のプロファイルにプロバティー値を設定するには、set サブコマンドとともに netcfg コマンドを使用します。このサブコマンドは、対話型モードまたはコマンド行モードで使用できます。プロバティー値をコマンド行モードで設定または変更する場合は、変更が永久ストレージでただちに確定されます。

set サブコマンドの構文は、次のとおりです。

```
netcfg set prop-name=value1[,value2...]
```

特定のプロバティー値を取得する必要がある場合は、netcfg get コマンドを使用します。詳細は、104 ページの「特定のプロバティーの値を取得する」を参照してくださ。

例4-9 プロバティー値を netcfg コマンド行モードで設定する

```
netcfg set mtu=1492
```

たとえば、net1 という名前のリンク NCU に mtu プロバティーを設定するには、次のコマンドを入力します。

```
$ netcfg "select ncp User; select ncu phys net1; set mtu=1492"
```

この例では、最上位のプロファイルを選択し、さらに変更された mtu プロバティー値が含まれる NCU を選択するために select サブコマンドを使用しています。

指定されたプロバティーに、コマンド行から同時に複数の値を設定できます。複数の値を設定する場合は、各値をコンマ(,)で区切る必要があります。指定されたプロバティーの各値はコンマも含まれている場合は、プロバティー値の一部であるコンマの前後に、バックラッシュ(\,)を付ける必要があります。単一の値のみが含まれるプロバティーのコンマは、区切り文字とは解釈されないため、前にバックラッシュを付ける必要はありません。

次の例では、NCP User で NCU の ip-version プロバティー値として myncu が設定されます。

```
$ netcfg "select ncp User; select ncu ip myncu; set ip-version=ipv4,ipv6"
```

例4-10 プロファイルのプロバティー値を対話形式で設定する

```
プロバティー値を対話形式で設定する場合は、まず現在のスコープでプロファイルを選択します。これにより、対話型セッションがプロファイルのスコープに移動します。このスコープから、変更するプロバティーが含まれるオブジェクトを選択できます。その後、選択したプロファイルが永久ストレージからメモリーに読み込まれます。このスコープでは、次の例に示すように、プロファイルまたはそのプロバティーを変更できます。
```

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化 - 2011年12月
例4-10 プロファイルのプロパティ値を対話形式で設定する (続き)

$ netcfg
netcfg> select ncp User
netcfg:ncp:User> select ncu ip iwk0
netcfg:ncp:User:ncu:iwk0> set ipv4-default-route = 129.174.7.366

次の例では、場所 foo の ipfilter config-file プロパティが設定されます。

$ netcfg
netcfg> list
NCPs: Automatic User
Locations: Automatic NoNet foo

netcfg> select loc foo
netcfg:loc:foo> list
LOC:foo
  activation-mode manual
  enabled false
  nameservices dns
dns-nameservice-configsrf dhcp
dns-nameservice-config-file */etc/nsswitch.dns*
netcfg:loc:foo> set ipfilter-config-file=/path/to/ipf-file
netcfg:loc:foo> list
LOC:foo
  activation-mode manual
  enabled false
  nameservices dns
dns-nameservice-configsrf dhcp
dns-nameservice-config-file */etc/nsswitch.dns*
ipfilter-config-file */path/to/ipf-file*
netcfg:loc:foo> end
Committed changes
netcfg> exit
Nothing to commit
$

次の例では、NCP User で NCU net0 の link-mtu プロパティが対話形式で変更されます。

$ netcfg
netcfg> select ncp User
netcfg:ncp:User> select ncu phys net0
netcfg:ncp:User:ncu:net0> list
NCU:net0
  type link
  class phys
  parent "User"
  enabled true
  activation-mode prioritized
  priority-mode exclusive
例4-10 プロファイルのプロパティー値を対話形式で設定する（続き）

priority-group 1
netcfg:ncp:User:ncu:net0> set link-mtu=5000
netcfg:ncp:User:ncu:net0> list
NCU:net0
type link
class phys
parent "User"
enabled true
activation-mode prioritized
priority-mode exclusive
priority-group 1
link-mtu 5000
netcfg:ncp:User:ncu:net0> commit
Committed changes
netcfg:ncp:User:ncu:net0> exit
Nothing to commit
$

プロファイル情報についてシステムに問い合わせる

現在のスコープまたは指定されたスコープで存在するプロファイル、プロパティ値のペア、およびリソースをすべて一覧表示するには、listサブコマンドとともにnetcfgコマンドを使用します。すべてのプロファイルに関する一般的な情報についてシステムに問い合わせたり、特定のプロファイルに関する具体的な情報を取得したりするには、listサブコマンドを使用します。listサブコマンドは、対話型モードまたはコマンド行モードで使用できます。

プロファイルおよびその現在の状態に関する情報を取得する必要がある場合は、list?サブコマンドとともにnetadm?コマンドを使用します。詳細は、116ページの「プロファイルの現在の状態を表示する」を参照してください。

システム上のプロファイルをすべて一覧表示する

netcfg list コマンドは、システム上のすべてのシステム定義およびユーザーディフィルのプロファイルを一覧表示します。オプションを付けないでlistサブコマンドを使用すると、システム上にあるすべての最上位プロファイルが表示されます。このコマンドでは、各プロファイルの状態は一覧表示されません。プロファイルおよびその状態（オンラインまたはオフライン）のリストを表示するには、netadm list コマンドを使用します。

システム上のすべての最上位プロファイルを一覧表示するには、次のコマンドを入力します。

$ netcfg list
NCPs:
    Automatic
プロファイル情報についてシステムに問い合わせる

User
Locations:
Automatic
NoNet
home
office
ENMs:
myvpn
testenm
WLANs:
workwifi
coffeeshop
homewifi

この例では、次のプロファイルが一覧表示されます。

- NCP
  システム定義のプロファイルである Automatic NCP と、User という名前のユーザー定義の NCP の 2 つの NCP が一覧表示されます。

- 場所
  2 つのシステム定義 (Automatic と NoNet) および 2 つのユーザー定義 (home と office) の 4 つのプロファイルが一覧表示されます。

- ENM
  インストールおよび構成された VPN アプリケーション用に 1 つの ENM、およびテスト用に 1 つの ENM の 2 つの ENM が一覧表示されます。

- WLAN
  勤務用に 1 つの WLAN、地元のコーヒーショップ用に 1 つの WLAN、およびユーザーの自宅の無線ネットワーク用に 1 つの WLAN の 3 つの WLAN が一覧表示されます。

注-作成、変更、または削除できるのは、ユーザー定義のプロファイルのみです。

特定のプロファイルのプロパティー値をすべて一覧表示する

指定されたプロファイルのすべてのプロパティー値を一覧表示するには、list サブコマンドとともに netcfg コマンドを使用します。

list サブコマンドの構文は、次のとおりです。

```
$ netcfg list [ object-type [ class ] object-name ]
```
例4-11  NCU のプロパティ値をすべて一覧表示する
たとえば、User NCP で IP NCU のすべてのプロパティ値を一覧表示するには、次のコマンドを入力します。

$ netcfg "select ncp User; list ncu ip net0"
NCU:net0
  type interface
  class ip
  parent "User"
  enabled true
  ip-version ipv4
  ipv4-addrsrc dhcp
  ipv6-addrsrc dhcp,autoconf

例4-12  ENM のプロパティ値をすべて一覧表示する
次の例では、myenm という名前の ENM のすべてのプロパティが一覧表示されます。

$ list enm myenm
ENM:myenm
  activation-mode manual
  enabled true
  start "/usr/local/bin/myenm start"
  stop "/bin/alt_stop"

この例では、list サブコマンドの出力に次の情報が表示されます。

- この ENM の activation-mode プロパティは manual に設定されています。
- ENM は有効になっています。
- FMRI を使用するのではなく、start および stop メソッドプロパティが指定されています。

特定のプロパティの値を取得する

get サブコマンドとともに netcfg コマンドを使用すると、指定されたプロパティの特定の値を取得できます。このサブコマンドは、対話型モードまたはコマンド行モードで使用できます。

get サブコマンドの構文は、次のとおりです。

netcfg get [-V] prop-name

User NCP の一部である myncu という名前の NCU の ip-version プロパティの値を取得するには、次のコマンドを入力します。例:

$ netcfg "select ncp User; select ncu ip myncu; get -V ip-version"
ipv4
プロファイル情報についてシステムに問い合わせる

-Vオプションとともにgetサブコマンドを使用すると、ここで示すようにプロパティ値だけが表示されます。

```
netcfg:ncp:User:ncu:net0> get -V activation-mode
manual
```
それ以外の場合、プロパティとその値の両方が表示されます。たとえば:

```
netcfg:ncp:User:ncu:net0> get activation-mode
activation-mode manual
```

▼ 単一プロパティ値を対話形式で取得する方法
この手順では、netcfg get コマンドを netcfg 対話型モードで使用して、単一プロパティ値を取得する方法について説明します。この特定の手順のいくつかの例では、User NCP で NCU の単一プロパティ値を取得する方法を示しています。これら
の例は、デモ目的でのみ使用されます。このコマンドの使用時に指定する情報は、取得を試みるプロファイ
ルおよびプロパティ値によって異なります。

プロファイルのプロパティ値をすべて表示する場合は、代わりに walkprop サブコマンドを使用できます。このサブコマンドでは、特定のプロファイルのすべてのプロパティが1つずつ調査されるため、プロファイルのプロパティの1つまたはすべてを変更できます。詳細は、106 ページの「walkpropサブコマンドを使用して、プロパティ値を対話形式で表示および変更する」を参照してください。

1 netcfg 対話型セッションを開始します。
   $ netcfg
   netcfg>

2 取得するプロパティ値が含まれるプロファイルまたは構成オブジェクトを選択します。
   netcfg> select object-type [ class ] object-name

注 - class パラメータは、NCP を選択している場合にのみ適用可能です。また、phys と ip の両方のクラスの NCU が同じ名前を共有する場合、class パラメータを指定する必要があります。ただし、NCU 名が一意の場合は、class パラメータは必要ありません。

たとえば、User NCP を選択するには、次のように入力します。

   netcfg> select User NCP

この例では、User NCP を選択することで、選択したオブジェクトのスコープに対話型セッションが移動します。
プロファイル情報についてシステムに問い合わせる

3 (省略可能) プロファイルのコンポーネントを表示します。

```
netcfg:ncp:User> list
NCUs:
  phys  net0
  ip    net0
```

4 取得するプロパティ値が含まれるオブジェクトを選択します。
次の例では、User NCP でリンク (phys) NCU net0 が選択されます。

```
netcfg:ncp:User> select ncu phys net0
```

NCU net0 を選択すると、該当オブジェクトのスコープに対話型セッションが移動し、NCU の現在のプロパティがメモリーから読み込まれます。

5 指定されたプロパティ値を取得します。

```
netcfg:ncp:User:ncu:net0> get property-value
```
たとえば、activation-mode プロパティの値を取得するには、次のように入力します。

```
netcfg:ncp:User:ncu:net0> get activation-mode
activation-mode manual
```

次の手順  この時点で、set サブコマンドを使用してプロパティに新しい値を設定したり、変更を行わずに対話型セッションを終了したりできます。対話型モードでプロパティ値を変更する場合は、commit または exit サブコマンドを使用して、変更を保存する必要があることに注意してください。netcfg 対話型モードでのプロパティ値の設定については、100 ページの「プロファイルのプロパティ値を設定および変更する」を参照してください。

walkprop サブコマンドを使用して、プロパティ値を対話形式で表示および変更する

walkprop サブコマンドを使用すると、プロファイルのプロパティを対話形式で表示できます。このサブコマンドでは、プロファイルのプロパティが1つずつ調査され、プロパティごとに名前と現在の値が表示されま。指定されたプロパティの現在の値を変更するために使用可能な対話型コマンドプロントも表示されます。複数の値が含まれるプロパティの区切り文字は、コンマ (,)です。指定されたプロパティの各値にカンマが含まれている場合は、前にパックスラッシュ (\)を付ける必要があります。単一の値のみが含まれるプロパティ内のコンマは、区切り文字ではないため、前にパックスラッシュを付ける必要はありません。

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
注 - walkpropサブコマンドは、対話型モードで使用する場合にのみ有効です。

例4-13 特定のプロファイルのプロパティ値を表示および変更する
次の例では、場所 foo の activation-mode プロパティを表示してから、walkprop サブコマンドを使用して変更します。walkpropサブコマンドを使用するときは、setサブコマンドを使用してプロパティ値を設定する必要がないことに注意してください。

```
$ netcfg
netcfg> select loc foo
netcfg:loc:foo> list
loc:foo
    activation-mode         manual
    enabled                 false
    nameservices            dns
    nameservices-config-file /etc/nsswitch.dns
    dns-nameservice-configsrc dhcp
    nfsv4-domain            "Central.oracle.com"

netcfg:loc:foo> walkprop
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-all
conditions> advertised-domain is oracle.com
nameservices (dns) [dns|files|nis|ldap]>
nameservices-config-file ("/etc/nsswitch.dns")>
dns-nameservice-configsrc (dhcp) [manual|dhcp]>
nfsv4-domain ("Central.oracle.com")>
ipfilter-config-file>
ipfilter-v6-config-file>
ipnat-config-file>
ippool-config-file>
ike-config-file>
ipsecpolicy-config-file>
netcfg:loc:foo> list
loc:foo
    activation-mode         conditional-all
    conditions              "advertised-domain is oracle.com"
    enabled                 false
    nameservices            dns
    nameservices-config-file /etc/nsswitch.dns
    dns-nameservice-configsrc dhcp
    nfsv4-domain            "Central.oracle.com"

netcfg:loc:foo> commit
Committed changes
netcfg:loc:foo> end
netcfg> exit
$
プロファイル構成をエクスポートおよび復元する

exportサブコマンドを使用すると、プロファイル構成を保存および復元できます。プロファイルのエクスポートは、同一のネットワーク構成が必要な複数のサーバーを保守する責任のあるシステム管理者にとって役立つことがあります。exportサブコマンドは、対話型モードまたはコマンド行モードで使用できます。また、コマンドをコマンドファイルモードで使用すると、ファイルをコマンドの出力として指定できます。

exportサブコマンドのコマンド構文は、次のとおりです。

```
$ netcfg export [ -d ] [ -f output-file ] [ object-type [ class ] object-name ]
```

注 - exportサブコマンドの -dおよび-fオプションは、相互に独立して使用できます。

例4-14 プロファイル構成をエクスポートする

この例では、exportサブコマンドを使用して、システムのプロファイル構成を画面に表示します。

```
$ netcfg
netcfg> export
create ncp "User"
create ncu ip "net2"
set ip-version=ipv4
set ipv4-addrsrc=dhcp
set ipv6-addrsrc=dhcp,autoconf
end
create ncu phys "net2"
set activation-mode=manual
set link-mtu=5000
end
create ncu phys "wpi2"
set activation-mode=prioritized
set priority-group=1
set priority-mode=exclusive
set link-mac-addr="13:10:73:4e:2"
set link-mtu=1500
end
end
create loc "test"
```

Oracle Solaris管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
例4-14 プロファイル構成をエクスポートする（続き）

```
set activation-mode=manual
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set nsfsv4-domain="domain1.oracle.com"
end
create loc "foo"
set activation-mode=conditional-all
set conditions="system-domain is oracle.com"
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set nsfsv4-domain="domain1.oracle.com"
end
create enm "myenm"
set activation-mode=conditional-all
set conditions="ip-address is-not-in-range 1.2.3.4"
set start="/my/start/script"
set stop="/my/stop/script"
end
create wlan "mywlan"
set priority=0
set bssids="0:13:10:73:4e:2"
end
netcfg> end
$```

例4-15 プロファイル構成をnetcfg対話型モードでエクスポートする
次の例では、-dオプションとともにexportサブコマンドを使用します。-aオプションを付けると、destroy -aコマンドがnetcfg export出力の先頭行として追加されます。

```
$ netcfg
netcfg> export -d
destroy -a
create ncp "User"
create ncu ip "net2"
set ip-version=ipv4
set ipv4-addrsrc=dhcp
set ipv6-addrsrc=dhcp,autoconf
end
create ncu phys "net2"
set activation-mode=manual
set link-mtu=5000
end
create ncu phys "wpi2"
set activation-mode=prioritized
set priority-group=1
set priority-mode=exclusive
set link-mac-addr="13:10:73:4e:2"
set link-mtu=1500
end
end
```

第4章 • NWAMプロファイルの構成(タスク) 109
例4-15 プロファイル構成をnetcfg対話型モードでエクスポートする（続き）

```
create loc "test"
set activation-mode=manual
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set nftsv4-domain="domain.oracle.com"
end
create loc "foo"
set activation-mode=conditional-all
set conditions="system-domain is oracle.com"
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set nftsv4-domain="domain.oracle.com"
end
create enm "myenm"
set activation-mode=conditional-all
set conditions="ip-address is-not-in-range 1.2.3.4"
set start="/my/start/script"
set stop="/my/stop/script"
end
create wlan "mywlan"
set priority=0
set bssids="0:13:10:73:4e:2"
end
netcfg> end
```

例4-16 プロファイル構成をnetcfgコマンドファイルモードでエクスポートする

次の例では、-fオプションともにnetcfg export コマンドを使用することによって、User NCPの構成情報がファイルに書き込まれます。-fオプションを付けると、出力がuser2という名前の新規ファイルに書き込まれます。-dオプションを付けると、destroy -a コマンドがnetcfg export 出力の先頭行として追加されます。

```
$ netcfg export -d -f user2 ncp User

$ ls -al
```

```
drwx------ 3 root root 4 Oct 14 10:53 .
drwxr-xr-x 37 root root 40 Oct 14 10:06 ..
-rw-r--r-- 1 root root 352 Oct 14 10:53 user2
```

```
$ cat user2
destroy -a
create ncp "User"
create ncu ip "net2"
set ip-version=ipv4
set ipv4-addresssrc=dhcp
set ipv6-addresssrc=dhcp,autoconf
end
create ncu phys "net2"
```

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
例4-16 プロファイル構成をnetconfigコマンドファイルモードでエクスポートする（続き）

```
set activation-mode=manual
set link-mtu=5000
end
create ncu phys "wpi2"
set activation-mode=prioritized
set priority-group=1
set priority-mode=exclusive
set link-mac-addr="13:10:73:4e:2"
set link-mtu=1500
end
end
create loc "test"
set activation-mode=manual
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set dns-nameservice-configsrc=dnsm
set natsv4-domain="domain.oracle.com"
end
create loc "foo"
set activation-mode=conditional-all
set conditions="system-domain is oracle.com"
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set dns-nameservice-configsrc=dnsm
set natsv4-domain="domain.oracle.com"
end
create enm "myenm"
set activation-mode=conditional-all
set conditions="ip-address is-not-in-range 1.2.3.4"
set start="/my/start/script"
set stop="/my/stop/script"
end
create wlan "mywlan"
set priority=0
set bssid="0:13:10:73:4e:2"
end
```

ユーザーオーバーのプロファイルを復元する
次のように、-fオプションとともにnetconfigコマンドを使用すると、ユーザー定義のプロファイルを復元できます。

```
$ netconfig [-f ] profile-name
```
たとえば:

```
$ netconfig -f user2
```
このコマンドでは、エクスポートされた構成が含まれるコマンドファイルが実行されます。
ネットワーク構成の管理

ネットワーク構成管理は、プロファイルベースで、2つのネットワーク構成モード（手動と自動）の間で切り替えることで管理されます。モード切り替えには、該当するNCPを有効にします。手動ネットワーク構成の場合は、DefaultFixed NCPを有効にします。自動(NWAM)ネットワーク構成の場合は、Automaticまたはユーザ一定義のNCPを有効にします。

▼ 自動ネットワーク構成モードから手動ネットワーク構成モードに切り替える方法

NWAM構成管理で現在サポートされていない高度なネットワーキング機能を使用している場合や、手動ネットワーク構成管理を使いたい場合は、次の手順で示すように、DefaultFixed NCPを有効にできます。

1 rootユーザーになります。

2 DefaultFixed NCPを有効にします。
   
   # netadm enable -p ncp DefaultFixed

3 network/physical:defaultサービスが再起動され、オンラインになっていることを確認します。
   
   # svcstatus -xv network/physical:default

   svc:/network/physical:default (physical network interface configuration)
   State: online since Fri Aug 26 16:19:18 2011
   See: man -M /usr/share/man -s 1M ipadm
   See: man -M /usr/share/man -s 5 nwam
   See: /var/svc/log/network-physical:default.log
   Impact: None.
   
   #

4 DefaultFixed NCPがアクティブであることを確認します。
   
   # netadm list
   netadm: DefaultFixed NCP is enabled;
   automatic network management is not available.
   'netadm list' is only supported when automatic network management is active.

注 - netadmコマンドがサポートされるのは、ネットワーク構成が自動モードのときだけです。したがって、手動モードでは、コマンドの出力はDefaultFixedプロファイールが有効であることを示すだけになります。システム内のその他のNCPに関する情報は出力されません。
手動ネットワーク構成モードから自動ネットワーク構成モードに切り替える方法

手動ネットワーク構成モードから自動ネットワーク構成モードに戻すには、使用するネットワーク構成プロファイルを有効にします。

1. rootユーザーになります。

2. NCPを有効にします(たとえば、Automatic)。
   
   # netadm enable -p ncp Automatic

3. network/physical:default サービスが再起動され、オンラインになっていることを確認します。
   
   # svcs -xv network/physical:default
svc:/network/physical:default (physical network interface configuration)
State: online since Fri Aug 26 16:19:18 2011
   
   See: man -M /usr/share/man -s 1M ipadm
   See: man -M /usr/share/man -s 5 nwam
   See: /var/svc/log/network-physical:default.log
Impact: None.

4. NCPおよびその他のNWAMプロファイルの状態をチェックします。
   
   # netadm list -x

<table>
<thead>
<tr>
<th>TYPE</th>
<th>PROFILE</th>
<th>STATE</th>
<th>AUXILIARY STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ncp</td>
<td>Automatic</td>
<td>online</td>
<td>active</td>
</tr>
<tr>
<td>ncu:phys</td>
<td>net0</td>
<td>online</td>
<td>interface/link is up</td>
</tr>
<tr>
<td>ncu:ip</td>
<td>net0</td>
<td>online</td>
<td>interface/link is up</td>
</tr>
<tr>
<td>ncu:phys</td>
<td>net1</td>
<td>offline</td>
<td>interface/link is down</td>
</tr>
<tr>
<td>ncu:ip</td>
<td>net1</td>
<td>offline</td>
<td>conditions for activation are unmet</td>
</tr>
<tr>
<td>ncp</td>
<td>User</td>
<td>disabled</td>
<td>disabled by administrator</td>
</tr>
<tr>
<td>loc</td>
<td>Automatic</td>
<td>online</td>
<td>active</td>
</tr>
<tr>
<td>loc</td>
<td>NoNet</td>
<td>offline</td>
<td>conditions for activation are unmet</td>
</tr>
</tbody>
</table>

#
NWAM プロファイルの管理（タスク）

この章では、netadm コマンドを使用して、NCP、場所、ENM、およびWLAN プロファイルを管理する方法について説明します。netadm コマンドを使用すると、NCP を構成する個々の構成オブジェクトである NCU を管理したり、NWAM GUI が存在しない場合に NWAM デーモン (nwamd) と対話したりすることもできます。netadm コマンドの使用については、netadm(1M) のマニュアルページを参照してください。

この章の内容は次のとおりです。

- 116 ページの「プロファイルの状態に関する情報を取得する」
- 118 ページの「プロファイルをアクティブまたは非アクティブにする」
- 121 ページの「無線スキャンを実行して、使用可能な無線ネットワークに接続する」
- 122 ページの「NWAM ネットワーク構成のトラブルシューティング」

netcfg コマンドを使用したプロファイルの作成およびプロパティの構成については、第4章「NWAM プロファイルの構成（タスク）」を参照してください。

デスクトップから NWAM GUI を使用して、NWAM 構成と対話する方法およびネットワーク構成を管理する方法については、第6章「NWAM グラフィカル ユーザーインターフェースについて」を参照してください。

NWAM の紹介については、第2章「NWAM の紹介」を参照してください。

すべての NWAM コンポーネントおよび NWAM 構成の詳細は、第3章「NWAM 構成と管理 (概要)」を参照してください。
プロファイルの状態に関する情報を取り出す

`list`サブコマンドとともに`netadm`コマンドを使用すると、システムで使用可能なすべてのプロファイルとその現在の状態を表示したり、特定のプロファイルとその状態を表示したりできます。

`list`サブコマンドの構文は、次のとおりです。

```
netadm list [ -p profile-type ] [ -c ncu-class ] [ profile-name ]
```

たとえば、システム上のすべてのプロファイルとその状態を表示するには、次のコマンドを入力します。

```
$ netadm list
```

<table>
<thead>
<tr>
<th>TYPE</th>
<th>PROFILE</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ncp</td>
<td>User</td>
<td>disabled</td>
</tr>
<tr>
<td>ncp</td>
<td>Automatic</td>
<td>online</td>
</tr>
<tr>
<td>ncu:ip</td>
<td>net1</td>
<td>offline</td>
</tr>
<tr>
<td>ncu:phys</td>
<td>net1</td>
<td>offline</td>
</tr>
<tr>
<td>ncu:ip</td>
<td>net0</td>
<td>online</td>
</tr>
<tr>
<td>ncu:phys</td>
<td>net0</td>
<td>online</td>
</tr>
<tr>
<td>loc</td>
<td>foo</td>
<td>disabled</td>
</tr>
<tr>
<td>loc</td>
<td>test</td>
<td>disabled</td>
</tr>
<tr>
<td>loc</td>
<td>NoNet</td>
<td>offline</td>
</tr>
<tr>
<td>loc</td>
<td>Automatic</td>
<td>online</td>
</tr>
</tbody>
</table>

この例では、システム上にあるすべてのシステム定義およびユーザーディスのプロファイルと、その現在の状態が表示されます。`list`サブコマンドでは、有効なNCPおよび特定のNCPを構成するすべてのNCUが表示されます。

プロファイルの現在の状態を表示する

コマンド構文にプロファイルタイプおよびNCUクラスを含めると、特定のプロファイルを識別できます。プロファイルタイプのみを指定した場合は、そのタイプのプロファイルがすべて表示されます。プロファイルを名前で指定した場合は、そのプロファイルの現在の状態が表示されます。プロファイル名が一意でない場合は、その名前のプロファイルがすべて一覧表示されます。

各プロファイルで取り扱う状態の値は、次のとおりです。

`disabled` まだ有効化されていないプロファイルが手動でアクティブ化されたことを示します。

`offline` まだアクティブ化されていないプロファイルが条件付きでアクティブ化またはシステムでアクティブ化されたことを示します。プロファイルの条件を満たしていなかったり、より限定された条件を満たした別のプロファイルがアクティブになっていなかったために、プロファイルがアクティブにならない場合があります。
プロファイルの状態に関する情報を取得する

注 - 場所プロファイルなどの、1つずつアクティブにする必要があるプロファイルタイプの場合、オフラインの状態がより頻繁に発生します。

online すでに条件を満たし、正常にアクティブになっているプロファイルが条件付きでアクティブ化またはシステムでアクティブ化されたことを示します。あるいは、ユーザーの要求で正常に有効になっているプロファイルが手動でアクティブ化されたことを示します。

maintenance プロファイルのアクティブ化が試行されたが、アクティブ化に失敗したことを示します。

initialized プロファイルは有効であるが、プロファイル上で操作が行われなかったことを示します。

uninitialized プロファイルがシステム上に存在しないことを示します。たとえば、物理リンクに対応するNCUがシステムから削除されると、この状態が発生する可能性があります。

例5-1 指定されたプロファイルの現在の状態を表示する
次の例では、名前で指定された Automatic NCP の現在の状態を一覧表示します。

$ netadm list Automatic
<table>
<thead>
<tr>
<th>TYPE</th>
<th>PROFILE</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ncp</td>
<td>Automatic</td>
<td>online</td>
</tr>
<tr>
<td>ncu:ip</td>
<td>net1</td>
<td>offline</td>
</tr>
<tr>
<td>ncu:phys</td>
<td>net1</td>
<td>offline</td>
</tr>
<tr>
<td>ncu:ip</td>
<td>net0</td>
<td>online</td>
</tr>
<tr>
<td>ncu:phys</td>
<td>net0</td>
<td>online</td>
</tr>
<tr>
<td>loc</td>
<td>Automatic</td>
<td>online</td>
</tr>
</tbody>
</table>

次の例では、-p オプションとともに list サブコマンドを使用し、システムに現在存在する場所をすべて表示します。

$ netadm list -p loc
<table>
<thead>
<tr>
<th>TYPE</th>
<th>PROFILE</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>loc</td>
<td>foo</td>
<td>disabled</td>
</tr>
<tr>
<td>loc</td>
<td>test</td>
<td>disabled</td>
</tr>
<tr>
<td>loc</td>
<td>NoNet</td>
<td>offline</td>
</tr>
<tr>
<td>loc</td>
<td>Automatic</td>
<td>online</td>
</tr>
</tbody>
</table>

次の例では、-c オプションとともに list サブコマンドを使用し、現在アクティブな NCP のインタフェース NCU をすべて表示します。
プロファイルをアクティブまたは非アクティブにする

例5-1 指定されたプロファイルの現在の状態を表示する 

```
$ netadm list -c ip
TYPE PROFILE STATE
ncu:ip net0 online
ncu:ip net1 disabled
```

補助的な状態の値

プロファイルの補助的な状態は、指定されたプロファイルがオンラインまたはオフライン（有効または無効）である原因について説明しています。補助的な状態の値を一覧表示するには、次の例に示すように、-xオプションとともにlistサブコマンドを使用します。

```
$ netadm list -x
TYPE PROFILE STATE AUXILIARY STATE
ncp Automatic disabled disabled by administrator
ncp User online active
ncu:phys nge0 online interface/link is up
ncu:ip nge0 online interface/link is up
ncu:phys nge1 offline interface/link is down
ncu:ip nge1 offline conditions for activation are unmet
loc Automatic offline conditions for activation are unmet
loc NoNet offline conditions for activation are unmet
loc office online active
```

補助的な状態の値は、プロファイルタイプによって異なります。補助的な状態の値の詳細は、nwamd(1M)のマニュアルページを参照してください。

プロファイルをアクティブまたは非アクティブにする

ユーザー定義のNCP、場所プロファイル、およびENMにはすべて、activation-modeプロパティーが存在します。各プロファイルで許容される値は、プロファイルのタイプで決まります。

プロファイルや構成オブジェクトを手動で有効または無効(アクティブまたは非アクティブ)にするには、netadm enableコマンドまたはnetadm disableコマンドを使用します。指定されたプロファイルのactivation-modeプロパティーがmanualに設定されている場合は、システム定義とユーザー定義の両方のプロファイルを有効および無効にすることができます。netcfgコマンドを使用してプロファイルを作成または変更すると、activation-modeプロファイルが設定されます。詳細は、58ページの「NWAMプロファイルをアクティブにする方法」を参照してください。

システムには常に、1つのアクティブなNCPと1つのアクティブな場所プロファイルが必要です。activation-modeをmanualにして別のNCPまたは場所を有効にする
プロファイルをアクティブまたは非アクティブにする

と、現在アクティブな NCP または場所プロファイルが暗黙的に非アクティブになります。activation-mode プロパティが manual に設定されている場合は、現在の場所も非アクティブにすることができます。その他の場所が使用できない場合、NWAM はシステム定義の場所（場所「Automatic」、または IP の構成に成功した場合は場所「NoNet」）のいずれかにに戻ります。条件付きおよびシステムの場所は、手動でアクティブにすることができます。つまり、明示的に無効にするまで、場所はアクティブのままです。この動作によって、条件付きの場所プロファイルを「常にオフ」の状態に切り替えることができます。条件付きの場所を無効にすると、システムは通常の条件下動作に戻ります。場所を手动で有効にした場合は、条件付きで有効化された場所の条件を満たしていても、場所は変更されません。

注 - システムで現在アクティブな NCP を明示的に無効にすることはできません。これにより、システムの基本ネットワーク接続を効率的にシャットダウンします。別の NCP を手動で有効にすると、NCP は暗黙的に無効になります。ただし、ENM のアクティブ化には制約があります。常にシステムで 0 個以上の ENM をアクティブにすることができます。したがって、1 つの ENM を有効または無効にしても、ほかの現在アクティブな ENM には影響しません。

個々の NCU を手動で有効または無効にすることもできます。指定された NCU は現在アクティブな NCP の一部にする必要があり、activation-mode プロパティを manual にする必要があることに注意してください。NCU クラスが指定されていない場合は、すべての NCU（その名前の 1 つのリンク NCU と 1 つのインタフェース NCU）がアクティブまたは非アクティブになります。

オブジェクトのアクティブ化と非アクティブ化は非同期的に実行されます。したがって、有効または無効にする要求に成功しても、操作（アクティブ化または非アクティブ化）には失敗する場合もあります。この種類の失敗はプロファイルの状態に反映され、プロファイルで最後に行われた操作に失敗したことを示す maintenance 状態に変わります。プロファイルの状態の表示については、119 ページの「プロファイルの状態に関する情報の取得」を参照してください。

例 5-2 プロファイルを有効にする

プロファイルを手動で有効にするための構文は、次のとおりです。

```bash
netadm enable [ -p profile-type ] [-c ncu-class ] profile-name
```

プロファイル名が一意でない場合（たとえば、システム上に同じ名前のプロファイルが複数存在するが、タイプは異なる場合）、プロファイルタイプも指定する必要があ ります。

第 5 章・NWAM プロファイルの管理 (タスク) 119
プロファイルをアクティブまたは非アクティブにする

例5-2 プロファイルを有効にする (続き)

-p オプションを使用すると、次のプロファイルタイプのいずれかを指定できます。

- ncp
- ncu
- loc
- enm

構成オブジェクトのタイプがncu の場合は、-c オプションを使用すると、NCU クラスを区別できます。-c オプションは、システム上に同じ名前の NCU が 2 つ存在するときに役立ちます。

-c オプションを使用する場合は、phys または ip のどちらかのクラスタイプを指定する必要があります。

次の例では、office という場所が有効になっています。

```
$ netadm enable -p loc office
```

ここで、profile-type は loc, profile-name は officeです。この例では、プロファイルタイプは場所であり、NCP ではないため、-c ncu-class オプションが使用されていないことに注意してください。

```
$ netadm enable -p ncp user
Enabling ncp 'User'
```

プロファイル名を指定する際に、netadm コマンドでは大文字と小文字が区別されないことに注意してください。

例5-3 プロファイルを無効にする

プロファイルを手動で無効にするための構文は、次のとおりです。

```
netadm disable [ -p profile-type ] [ -c ncu-class ] profile-name
```

プロファイル名が一致しない場合は、プロファイルタイプも入力する必要があります。

-p オプションを使用すると、次のプロファイルまたはオブジェクト型のいずれかを指定できます。

- ncp
- ncu
- loc
- enm
例5-3 プロファイルを無効にする (続き)
構成オブジェクトのタイプがncuの場合は、-cオプションを使用して、NCUクラスを区別する必要もあります。
NCUクラスには、physとipのいずれかを指定する必要があります。
たとえば、net1という名前のリンクNCUを手動で無効にするには、次のコマンドを入力します。

$ netadm disable -p ncu -c phys net1
ここでは、profile-typeはncu、ncu-classはphys、profile-nameはnet1です。この例では、構成オブジェクトがNCUであるため、-cncu-classオプションが使用されていることに注意してください。

例5-4 プロファイルの切り替え
アクティブなNCPを変更して、手動構成を有効にするには、次のコマンドを入力します。

$ netadm enable -p ncp DefaultFixed
同様に、AutomaticNCPで自動(NWAM)構成を有効にするには、次のコマンドを入力します。

$ netadm enable -p ncp Automatic
netadmの詳細は、netadm(1M)のマニュアルページを参照してください。

無線スキャンを実行して、使用可能な無線ネットワークに接続する
netadmコマンドを使用すると、使用可能な無線ネットワークをスキャンして接続できます。
無線リンクをスキャンして使用可能な無線ネットワークのリストを取得するには、netadmscan-wifilink-nameコマンドを使用します。

link-nameとして指定されたリンク上で、スキャン結果から無線ネットワークを選択して接続するには、netadmselct-wifilink-nameコマンドを使用します。select-wifilink-nameサブコマンドでは、必要に応じてWiFiの選択、キーおよびキースロットを求めるプロンプトが表示されます。
注 - netadm select-wifi コマンドを使用する前に、キーを作成しておく必要があります。

netadm scan-wifi link-name コマンドを使用すると、使用可能な無線ネットワークを検索するための後続のネットワークスキャンをトリガーすることもできます。新規スキャン結果が既存のスキャン結果と同一である場合は、後続のスキャンでスキャンイベントがトリガーされない可能性があることに注意してください。nwamdデーモンは、最後のスキャン以降にデータが変更されたかどうかに関係なく、スキャンを実行します。

次の例では、netadm scan-wifi コマンドを使用して、無線リンク net1 のスキャンを実行します。次に、netadm select-wifi コマンドを使用して、選択する無線ネットワークのリストを表示します。表示されるリストは、以前に net1 で実行されたスキャンの結果に基づきます。

```
$ netadm scan-wifi net1
1: ESSID home BSSID 0:b:e:85:26:c0
2: ESSID neighbor1 BSSID 0:b:e:49:2f:80
3: ESSID testing BSSID 0:40:96:29:e9:d8
4: Other
```
Choose WLAN to connect to [1-4]: 1
```
WLAN でキーが必要な場合、WEP が指定されていれば、キーおよびキーを入力するプロンプトが表示されます。たとえば:

Enter WLAN key for ESSID home: mywlankey
Enter key slot [1-4]: 1
```

NWAM ネットワーク構成のトラブルシューティング

この節では、NWAM ネットワーク構成に関する問題のトラブルシューティングを行う方法について説明します。

すべてのネットワーク接続の現在の状態を監視する

show-events サブコマンドとともに netadm コマンドを使用すると、NWAM デーモン nwamd で監視されるイベントを待機して表示できます。このサブコマンド
は、NWAMで構成されたプロファイルおよび構成オブジェクトの構成プロセスに関連するイベントについて、役立つ情報を提供します。

netadm show-events コマンドの構文は、次のとおりです。

netadm show-events [-v]

次の例では、-vオプションとともにnwam show-eventsコマンドを使用して、イベントを詳細モードで表示します。

$ netadm show-events -v

<table>
<thead>
<tr>
<th>EVENT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINK_STATE</td>
<td>net0 -&gt; state down</td>
</tr>
<tr>
<td>OBJECT_STATE</td>
<td>ncu link:net0 -&gt; state online*, interface/link is down</td>
</tr>
<tr>
<td>OBJECT_STATE</td>
<td>ncu link:net0 -&gt; state offline, interface/link is down</td>
</tr>
<tr>
<td>OBJECT_STATE</td>
<td>ncu interface:net0 -&gt; state online*, conditions for act</td>
</tr>
<tr>
<td>IF_STATE</td>
<td>net0 -&gt; state (0) flags 2004801</td>
</tr>
<tr>
<td>IF_STATE</td>
<td>net0 -&gt; state (0) flags 2004800</td>
</tr>
<tr>
<td>IF_STATE</td>
<td>net0 -&gt; state (0) flags 1004803</td>
</tr>
<tr>
<td>IF_STATE</td>
<td>net0 -&gt; state index 4 flags 0x0 address fe80::214:4fff:</td>
</tr>
<tr>
<td>IF_STATE</td>
<td>net0 -&gt; state index 4 flags 0x0 address 129.156.235.229</td>
</tr>
<tr>
<td>IF_STATE</td>
<td>net0 -&gt; state index 4 flags 0x0 address 129.156.235.229</td>
</tr>
<tr>
<td>IF_STATE</td>
<td>net0 -&gt; state (0) flags 1004803</td>
</tr>
<tr>
<td>IF_STATE</td>
<td>net0 -&gt; state (0) flags 1004802</td>
</tr>
<tr>
<td>IF_STATE</td>
<td>net0 -&gt; state (0) flags 1004803</td>
</tr>
<tr>
<td>IF_STATE</td>
<td>net0 -&gt; state (0) flags 1004802</td>
</tr>
</tbody>
</table>

ネットワークインタフェース構成に関する問題のトラブルシューティング

netadm list -xコマンドは、ネットワークインタフェースが正しく構成されない場合の原因を特定する際に役立ちます。このコマンドは、NWAMで構成されたさまざまなエンティティー、エンティティーの現在の状態、およびエンティティーがその状態である理由を表示します。

たとえば、ケーブルが抜かれている場合に、netadm list -x コマンドを使用すると、リンク状態がオフラインであるかどうか、およびその理由（「link is down」など）を特定できます。同じく、重複アドレスが検出された場合は、netadm list -x コマンドの出力に、物理リンクがオンライン（起動中）であるが、IPインタフェースは保守状態であることが表示されます。この例では、表示される理由は「アドレスの重複が検出されました」です。

netadm list -x コマンドの出力例は、次のとおりです。

$ netadm list -x

<table>
<thead>
<tr>
<th>TYPE</th>
<th>PROFILE</th>
<th>STATE</th>
<th>AUXILIARY STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ncp</td>
<td>Automatic</td>
<td>online</td>
<td>active</td>
</tr>
</tbody>
</table>

第5章・NWAMプロファイルの管理(タスク) 123
### NWAMネットワーク構成のトラブルシューティング

<table>
<thead>
<tr>
<th>Interface</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ncu:phys net0</td>
<td>offline</td>
<td>interface/link is down</td>
</tr>
<tr>
<td>ncu:ip net0</td>
<td>offline</td>
<td>conditions for activation are unmet</td>
</tr>
<tr>
<td>ncu:phys net1</td>
<td>offline*</td>
<td>need WiFi network selection</td>
</tr>
<tr>
<td>ncu:ip net1</td>
<td>offline</td>
<td>conditions for activation are unmet</td>
</tr>
<tr>
<td>ncp User</td>
<td>disabled</td>
<td>disabled by administrator</td>
</tr>
<tr>
<td>loc Automatic</td>
<td>offline</td>
<td>conditions for activation are unmet</td>
</tr>
<tr>
<td>loc NoNet</td>
<td>online</td>
<td>active</td>
</tr>
<tr>
<td>loc office</td>
<td>offline</td>
<td>conditions for activation are unmet</td>
</tr>
</tbody>
</table>

リンクまたはインターフェースがオフラインになっている原因が特定されると、問題の修正に進むことができます。IPアドレスが重複している場合は、netcfgコマンドを使用して、指定されたインターフェースに割り当てられた静的IPアドレスを変更する必要があります。手順については、「プロファイルのプロパティ値を設定および変更する」を参照してください。変更を確定したら、netadm list -xコマンドを再度実行して、インターフェースが正しく構成され、状態がonlineと表示されることをチェックします。

インターフェースが正しく構成されない場合の原因のもう1つの例として、使用可能な既知のWLANが存在しないことが挙げられます。この場合、WiFiリンクの状態がofflineと表示され、その理由として「need wifi selection」と表示されます。また、WiFiの選択は行われたが、キーが必要である場合は、その理由として「need wifi key」と表示されます。
NWAM グラフィカルユーザーインタフェースについて

この章では、NWAM グラフィカルユーザーインタフェース (GUI) を紹介し、NWAM GUI を構成するコンポーネントについて説明します。デスクトップからの NWAM との対話、ネットワーク接続の制御、無線ネットワークの追加、およびネットワークプロファイルの作成と管理のための基本手順についても、この章で説明します。

GUI を使用してネットワークを排他的に管理するためのステップ・バイ・ステップの手順については、この章では説明しません。詳細な手順については、オンラインヘルプを参照してください。オンラインヘルプには、デスクトップのパネル通知領域に常に表示される「ネットワークステータス」アイコンを右クリックすればアクセスできます。GUI のリンクをクリックすると、各トピックに関する詳細な情報を提供するオンラインヘルプのページにアクセスできます。また、テキストに表示されるリンクをクリックするか、サイド区画のさまざまなトピックをクリックして、オンラインヘルプをナビゲートすることもできます。

この章の内容は次のとおりです。
- 125 ページの「NWAM グラフィカルユーザーインタフェースの紹介」
- 128 ページの「NWAM GUI の機能コンポーネント」
- 131 ページの「デスクトップから NWAM と対話する」
- 135 ページの「お気に入りの無線ネットワークに加入して管理する」
- 137 ページの「ネットワークプロファイルの管理」
- 144 ページの「場所を作成および管理する」
- 147 ページの「外部ネットワーク識別子について」

NWAM グラフィカルユーザーインタフェースの紹介

NWAM グラフィカルユーザーインタフェース (GUI) は、NWAM コマンド行ユーザーインターフェースのグラフィカル版です。NWAM GUI を使用すると、デスクトップでネットワークのステータスを表示して監視したり、NWAM と対話して Ethernet および無線の構成を管理したりできます。さらに、起動時の有線または無線ネットワークへの接続や、新しい有線または無線ネットワークの構成などの、さま
さまざまなネットワークタスクをデスクトップから実行できます。NWAM GUI を使用すると、場所（つまり、システム全体にわたるネットワーク構成の複雑なタスクを単純化するプロファイル）を作成および管理することもできます。GUI コンポーネントには、現在のネットワーク接続のステータスに関する情報、およびネットワーク環境全体の状況に関する情報を表示する機能があります。

NWAM GUI の基本的な機能は次のとおりです。

- ネットワークステータスの通知
- ホットプラグイベントの検出
- ネットワークプロファイルの作成と管理
- 無線ネットワークの管理

NWAM GUI は、目的のプロパティ値をプロファイルの形式でシステム上に格納することによって、ネットワーク構成を NWAM CLI と同じ方法で管理します。NWAM サービスは、現在のネットワーク状況に基づいて、特定の時間にアクティブにするプロファイルを決定し、最適なプロファイルをアクティビにします。

**デスクトップから NWAM GUI にアクセスする**

NWAM GUI は、デスクトップパネルに継続的に表示される「ネットワークステータス」通知アイコン、および「システム」⇒「管理」メニューと通知アイコンの右クリックのどちらからでもアクセス可能なネットワークの構成ダイアログの 2 つのコンポーネントで構成されています。NWAM GUI は、電源管理アイコンやプリンタアイコンなどの継続的なステータス通知アイコンが付いたその他のアプリケーションと同様に動作します。これらのアプリケーションでは、右クリック（コンテキスト）メニューにアクセスするか、アイコンまたは各種設定メニューからアクセス可能な構成ダイアログを使用することによって、特定のタスクを実行できます。

パネルアイコンは、もっとも頻繁に NWAM と接続するポイントです。アイコンには、有線と無線のどちらのネットワークに現在接続しているのが表示されます。アイコンの上にマウスポインタを重ねると、現在アクティブな NCP や場所プロファイルなどの追加情報がツールチップに表示されます。アイコンを右クリックすると、別の無線ネットワークに接続するなどの、システムの基本的なネットワーク構成を変更できます。

パネルアイコンをクリック（左クリック）すると、「ネットワーク設定」ダイアログが開きます。このダイアログは、「システム」⇒「管理」メニューからでも開くことができます。ここでは、静的 IPv4 および IPv6 アドレスの定義、接続優先度の設定、外部ネットワーク修飾子 (ENM) の管理、さまざまな場所で使用するためのネットワーク設定グループの作成などの、より詳細なネットワーク構成を実行できます。
NWAM CLI と NWAM GUI の相違点

NWAM を使用したネットワーク構成は、CLI と GUI のどちらかを使用して管理できます。どちらのユーザーインタフェースを使用しても、ネットワーク構成の管理、およびNWAM構成との対話が可能です。特定のタスクを実行する際に、CLI とGUIの使用のどちらを選択するのかは、そのタスクや特定の状況によって異なります。タスクによっては、NWAM GUIを使用することが、もっとも論理的な選択であることもあります。例として、現在アクティブなネットワーク接続のステータスのチェックや、起動時に接続する無線ネットワークの選択が挙げられます。これらのタスクは、デスクトップからGUI経由でNWAMと直接対話するほうが、より簡単かつ迅速に実行できます。新規ENMの起動および停止方法としてスクリプトを指定するなどの、より複雑なタスクでは、コマンド行モードでの作業を選択できます。

CLI と GUIは基本的には同じですが、次の相違点があることに注意してください。

- 機能の相違点

GUIには、デスクトップからNWAMと対話して、ネットワーク接続をチェックできる機能が備わっています。ネットワークのステータスに関する情報を取得する方法は、GUIとCLIユーティリティ間で若干異なります。GUIコンポーネントを使用する場合は、通知が発生するとデスクトップに表示されます。コマンド行ユーティリティーを使用する場合には、NWAMイベントが発生したときにはnetadm show-eventsコマンドを使用して監視を行うことができます。詳細は、122 ページの「すべてのネットワーク接続の現在の状態を監視する」を参照してください。

また、GUIを使用してネットワークのステータスに関する情報を取得するには、デスクトップに表示される「ネットワークステータス」通知アイコンを視覚的にチェックしたり、マウスポインタを重ねたり、クリックしたりします。コマンド行からネットワークのステータスに関する情報を取得するには、listサブコマンドとともにnetadmコマンドを使用します。このコマンドの出力には、システムで構成されている各ネットワークオブジェクトの基本的な状態に関する情報が含まれます。ただし、GUIでは、接続されている無線ネットワークの種類やネットワーク接続のIPアドレスなどの、ネットワークステータスに関するより完全な情報や詳細が提供されます。

CLIを使用して実行できるコマンドの一部は、GUIを使用しては実行できません。たとえば、GUIコンポーネントを使用してプロファイル構成をエクスポートすることはできません。プロファイル構成をエクスポートするには、netcfg exportコマンドを使用します。詳細は、108 ページの「プロファイル構成をエクスポートおよび復元する」を参照してください。

- コンポーネント名と用途の使用上の相違点

GUIでは、ネットワーク構成プロファイル(NCP)はネットワークプロファイルと同じです。CLIでネットワーク構成ユニーク(NCU)と呼ばれているものは、GUIではネットワーク接続と呼ばれます。
NWAM GUI の機能コンポーネント

NWAM GUI には、CLI を使用して実行できるタスクと実質的に同じタスクを実現するために使用される複数の機能コンポーネントがあります。表 6-1 では、これらの各コンポーネントについて説明します。一部、さまざまな方法でアクセスしたり、開いたりできるダイアログもあることに注意してください。また、一部のダイアログでは、ダイアログにアクセスする方法に応じて異なる情報が表示されます。このような相違点に関する具体的な情報については、この章を通じて関連する節で述べられ、オンラインヘルプでも詳細に説明しています。

表 6-1 NWAM GUI の主要なコンポーネント

<table>
<thead>
<tr>
<th>コンポーネント</th>
<th>機能</th>
<th>アクセス方法</th>
</tr>
</thead>
<tbody>
<tr>
<td>「ネットワークステータス」通知アイコン</td>
<td>ネットワークのステータスを表示する方法およびデスクトップから NWAM と対話する方法。このアイコンには、GUI を使用してネットワーク構成を作成および管理するためにアクセス可能な コンテキストメニューが付いています。</td>
<td>常にデスクトップパネルの通知領域に表示されているアイコンを参照します。</td>
</tr>
<tr>
<td></td>
<td></td>
<td>アイコンの上にマウスポインタを重ねると、現在のネットワークステータスに関する情報を提供するツールチップが表示されます。</td>
</tr>
<tr>
<td></td>
<td></td>
<td>アイコンをクリックすると、「ネットワーク設定」ダイアログが表示されます。</td>
</tr>
<tr>
<td></td>
<td></td>
<td>アイコンを右クリックすると、コンテキストメニューが開きます。</td>
</tr>
</tbody>
</table>
NWAM GUIの機能コンポーネント

表6-1 NWAM GUIの主要なコンポーネント（続き）

<table>
<thead>
<tr>
<th>コンポーネント</th>
<th>機能</th>
<th>アクセス方法</th>
</tr>
</thead>
<tbody>
<tr>
<td>「ネットワーク設定」ダイアログ</td>
<td>主要な2種類のネットワークプロファイルダイブ（システム定義の「自動」プロファイルと複数のユーザー定義のネットワークプロファイル）をアクティブ化および管理する方法。 「自動」およびユーザー定義のネットワークプロファイルは、ネットワークインタフェースごとにネットワーク構成を管理します。このダイアログは、ネットワークインタフェースごとにIPv4およびIPv6アドレスを構成して、お気に入りの無線ネットワークを管理する際にも使用されます。</td>
<td>■ デスクトップ上の「ネットワークステータス」通知アイコンをクリックします。 ■ デスクトップパネルのメインメニューから、「システム」→「管理」→「ネットワーク」の順に選択します。 ■ 「ネットワークステータス」通知アイコンの右クリックメニューから「ネットワークの場所」を選択します。 ■ また、「ネットワーク設定」ダイアログの「接続ステータス」ビューから、「場所」ボタンをクリックします。</td>
</tr>
</tbody>
</table>

「ネットワークの場所」ダイアログ | システム定義およびユーザー定義の場所プロファイルのプロバディーを作成し、有効化、および管理する方法。場にネームサービスやファイアウォールの設定などのネットワーク構成の特定要素を指定します（必要に応じて、同時に適用されます）。 | ■ 「ネットワークステータス」通知アイコンの右クリックメニューから「ネットワークの場所」を選択します。 ■ または、「ネットワーク設定」ダイアログの「接続ステータス」ビューから、「場所」ボタンをクリックします。 |
### 表6-1  NWAM GUIの主要なコンポーネント （続き）

<table>
<thead>
<tr>
<th>コンポーネント</th>
<th>機能</th>
<th>アクセス方法</th>
</tr>
</thead>
<tbody>
<tr>
<td>「無線ネットワークの加入」ダイアログ</td>
<td>無線ネットワークに加入する方法およびお気に入りネットワークのリストを管理する方法。注-このダイアログは、無線ネットワークの追加を試みて、そのネットワークに関する詳細情報が必要な場合に自動的に開きます。</td>
<td>通知アイコンの右クリックメニューから「リストがない無線ネットワークの加入」オプションを選択します。</td>
</tr>
<tr>
<td>「無線チャンネル」ダイアログ</td>
<td>無線ネットワークを選択する方法および接続する方法。</td>
<td>次の通知メッセージをクリックします。「インテリジェントがESSIDから切断しました。その他の使用可能なネットワークを表示するには、このメッセージをクリックしてください。」注-このダイアログは、加入可能な無線ネットワークの選択肢がある場合に、必ず自動的に開きます。</td>
</tr>
<tr>
<td>「ネットワーク修飾者」ダイアログ</td>
<td>ネットワーク構成の作成や変更が可能な外部ネットワーク修飾子アプリケーションを追加する方法。</td>
<td>「ネットワーク設定」ダイアログの「接続ステータス」ビューで「修飾者」ボタンをクリックします。「ネットワーク修飾者」通知アイコンを右クリックしてから、「ネットワーク修飾者の設定」メニュー項目を選択します。</td>
</tr>
</tbody>
</table>
デスクトップからNWAMと対話する

常にデスクトップバブルの通知領域に表示されている「ネットワークステータス」通知アイコンは、ネットワークのステータスを表示したり、自動ネットワーク構成のプロセスと対話したりするための主要な方法です。「ネットワークステータス」通知アイコンは、ネットワークに関する情報メッセージが表示される場所でもあります。アイコンのコンテキスト(右クリック)メニューを使用すると、基本的なネットワーク機能をすばやくアクセスできます。このアイコンの外観は、ネットワークの全体の狀態を示します。

ネットワーク接続のステータスの確認

ネットワークに関する重要な情報は、デスクトップのバブル通知領域に表示される「ネットワークステータス」通知アイコンをチェックすることで、すばやく確認できます。「ネットワークステータス」通知アイコンは、現在有効なネットワーク接続のステータスを参照し、NWAMサービスを操作するための主な手段です。アイコンの表示は、現在有効なネットワーク接続のステータスに応じて変化します。現在有効なネットワーク接続に関する情報は表示する別の方法として、「ネットワークステータス」通知アイコン上にマウスのポインタを重ねる方法もあります。通知アイコンのコンテキストメニューにアクセスするには、アイコンを右クリックします。このコンテキストメニューから、現在有効なネットワークインタフェースを変更し、接続先の無線ネットワーク(存在する場合)に関する詳細情報を参照できます。

注－「ネットワークステータス」通知アイコンは、NWAMを使用してネットワークを自動的設定している場合のみデスクトップに表示されます。

次の表は、システムで有効なネットワーク接続のステータスを反映して変化する「ネットワークステータス」アイコンの表示を示しています。
デスクトップからNWAMと対話する

<table>
<thead>
<tr>
<th>アイコン</th>
<th>状態</th>
<th>説明</th>
</tr>
</thead>
</table>
| ![アイコン1] | すべてオンライン (有線) | 有効なネットワークプロファイル内にある手動で有効になったすべての接続がオンラインであり、かつ有効なプロファイルグループ内の必要な数の接続（このようなグループが存在する場合）がオンラインであることを示します。「必要な数」は次のとおりです。  
- そのグループの優先度タイプが「排他的」である場合は、1つの接続  
- そのグループの優先度タイプが「共有」である場合は、1つ以上の接続  
- そのグループの優先度タイプが「すべて」である場合は、グループ内のすべての接続 |
| ![アイコン2] | すべてオンライン (無線) | 有効なネットワークプロファイル内にある手動で有効になったすべての接続がオンラインであり、かつ有効なプロファイルグループ内の必要な数の接続（このようなグループが存在する場合）がオンラインであることを示します。「必要な数」は、「すべてオンライン (有線)」ステータスで説明した数と同じです。

少なくとも1つのオンライン接続が無線であることに注意してください。 |
| ![アイコン3] | 部分的にオンライン (有線) | 1つ以上の手動で有効化された接続または優先グループ接続がオフラインであり、ステータスが「すべてオンライン」でなくなったことを示します。この例では、少なくとも1つの有線接続がオンラインです。
「ネットワークステータス」通知アイコンはまた、無線接続でユーザー入力が保留されている場合も「部分的にオンライン」として表示されます。たとえば、使用可能な無線ネットワークを選択している場合や、無線ネットワークのパスワードを提供している場合などがあります。

「ネットワークステータス」通知アイコンはまた、無線接続でユーザー入力が保留されている場合も「部分的にオンライン」として表示されます。たとえば、使用可能な無線ネットワークを選択している場合や、無線ネットワークのパスワードを提供している場合などがあります。 |
| ![アイコン4] | オフライン (有線) | NWAMサービスが無効になっているか、または保守モードにあることを表示します。 |

Oracle Solaris管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
有効なネットワーク接続に関する詳細を表示する方法

1 「ネットワーク設定」ダイアログを開いて、必要に応じてドロップダウンリストから「接続ステータス」を選択します。
   「ネットワーク設定」ダイアログは、次の場合のいずれかで開くことができます。
   - デスクトップパネルの「ネットワークステータス」通知アイコンをクリックします。
   - デスクトップパネルのメインメニューから、「システム」→「管理」→「ネットワーク」の順に選択します。

2 「ネットワークステータス」通知アイコンを右クリックしてメニューを開いてから、「ネットワーク設定」を選択します。
   有線ネットワーク接続の場合は、IPアドレス、信号の強さ、接続速度、接続ステータス、およびセキュリティーのタイプが表示されます。

特定のネットワーク接続の追加プロパティーを表示または編集するには、リスト内の接続をダブルクリックするか、ダイアログの一番上にある「表示」ドロップダウンメニューから接続を選択します。

デスクトップからネットワーク接続を制御する

デフォルトでは、NWAMは常にネットワーク接続の維持を試みます。有線ネットワーク接続に失敗すると、お気に入りの無線ネットワークのいずれかへの接続が試みられます。その試みにも失敗すると、アクセス権を持つ、その他の使用可能な無線ネットワークへの接続が試みられます。

また、必要に応じて、有線ネットワークと無線ネットワークを手動で切り替えることもできます。

注 - すべての接続タイプで、接続動作は現在のセッションのみに設定されます。システムリブートまたは接続解除すると、有効なネットワークプロファイアルで定義されている優先順に従ってネットワーク接続の確立が試みられます。

NWAMを次のようにすることで、ネットワーク接続をデスクトップから制御できます。

- デフォルトの接続優先順の変更
  デフォルトでは、すべての有線ネットワークがすべての無線ネットワーク接続よりも優先されます。つまり、無線ネットワーク接続が試みられるのは、有線接続を確立できない場合に限ります。現在の場所で複数の無線ネットワークが利用で
デスクトップからNWAMと対話する

きる場合は、加入するネットワークの選択が求められます。この動作は「自動」ネットワークプロファイルで定義され、デフォルトでアクティブになっています。別の動作を強制するには、別のネットワークプロファイルを作成してアクティブにする必要があります。

■ 有線ネットワークからの無線ネットワークへの切り替え
「自動」ネットワークプロファイルが有効な場合は、有効になっているすべての有線インタフェースからネットワークケーブルをすべて切り離します。
デフォルトでは、お気に入りの無線ネットワークのいずれかが使用可能な場合では、お気に入りリストに表示される順序での加入が試みられます。それ以外の場合は、「無線チューニング」ダイアログが表示されます。このダイアログでは、加入するネットワークを選択できます。

注 - 無線ネットワークに加入する方法は、「接続プロパティ」ビューの「無線」タブで変更できます。

「自動」ネットワークプロファイル以外のネットワークプロファイルが有効な場合、無線ネットワークに切り替えるために使用する方法は、そのネットワークプロフィールの定義によって異なります。

次のいずれかの方法から選択します。

■ 「ネットワークステータス」通知アイコンの「接続」サブメニューを使用して、有線接続を無効にしてから、無線接続をアクティブにします。この方法が使用可能ならば、両方の接続の起動型が「手動」である場合に限ることに注意してください。

■ 有線接続をアクティブにしてから、必要に応じてその他の接続を無効にするには、有効なネットワークプロファイルを編集します。

無線接続が確立されると、通知メッセージが表示されます。

■ 無線ネットワークからの有線ネットワークへの切り替え
「自動」ネットワークプロファイルが有効な場合は、使用可能な有線インタフェースにネットワークケーブルを接続します。
「自動」ネットワークプロファイル以外のネットワークプロファイルが有効な場合、有線ネットワークに切り替えるために使用する方法は、そのネットワークプロファイルの定義によって異なります。
お気に入りの無線ネットワークに加入して管理する

NWAMのデフォルトでは、無線ネットワーク接続が有効な場合、接続が一覧表示される優先度の順で、お気に入りリストで使用可能なネットワークへの接続が要求なしで試みられます。使用可能なお気に入りのネットワークがない場合は、「無線チューニング」ダイアログが開きます。このダイアログでは、加入する無線ネットワークを選択できます。

「ネットワーク設定」ダイアログの「接続プロパティ」ビューの「無線」タブで、無線ネットワークへの接続が試行される方法を変更することもできます。必要に応じて、「ネットワークステータス」通知アイコンの右クリックメニューにアクセスして、別な無線ネットワークに手動で接続できます。

ヒント - 「ネットワーク設定」ダイアログを使用して、選択したネットワークの「接続プロパティ」ビューにアクセスできます。このダイアログには、「表示」というラベルが付いたドロップダウンリストが付いています。このリストでは、特定のネットワークのビューを切り替えることができます。各ビューでは、実行可能なさまざまなタスクが、選択したネットワークに関する、該当ビューに固有の情報が提供されます。

次のビューは、システム上にある各ネットワークプロファイルのネットワーク接続ごとに存在します。

- 接続ステータス
- ネットワークプロファイル
- 接続プロパティ

ネットワークプロファイルの操作の詳細（「ネットワーク設定」ダイアログの説明を含む）は、137ページの「ネットワークプロファイルの管理」を参照してください。
お気に入りの無線ネットワークに加える方法

1. 別の無線ネットワークに手動で接続するには、次のいずれかを行うことができます。
   - 「ネットワークステータス」通知アイコンの右クリックメニューから、使用可能な無線ネットワークを選択します。
   - 「ネットワークステータス」通知のアイコンメニューから「リスト以外の無線ネットワークに加える」オプションを選択します。
     リスト以外の無線ネットワークとは、そのネットワーク名をブロードキャストしないように構成されているけれども、引き続き加入が可能な無線ネットワークのことです。
   - 「無線チューニング」ダイアログから、使用可能な無線ネットワークを選択します。加入可能な無線ネットワークの選択肢がある場合は、このダイアログが自動的に表示されます。

![無線ネットワークの検出結果](image)

2. 「無線ネットワークの加入」ダイアログが開かれた後、選択した無線ネットワークに関するすべての必要情報を指定してください。
   指定する必要のある情報の詳細は、NWAM GUI のオンラインヘルプを参照してください。
お気に入りのネットワークを管理する

デフォルトでは、はじめて無線ネットワークに加入すると、「無線ネットワークに加入」ダイアログに「接続が成功したらお気に入りのネットワークのリストに追加する」という名前のチェックボックスが表示されます。

- お気に入りリストに無線ネットワークを追加するには、接続に成功したら、このボックスを選択します。お気に入りリストにネットワークを追加しない場合は、チェックを解除します。チェックはデフォルトで選択されています。
- 現在使用できない、または現在お気に入りリストにネットワーク名をブロードキャストしていない無線ネットワークを追加するには、「接続プロパティーピュー」の「無線」タブを移動してから、「追加」ボタンをクリックします。ネットワークを追加するには、ネットワーク名、セキュリティータイプ、およびセキュリティキーを知っている必要があります。

ネットワークプロファイアルの管理

NWAM GUIを使用する場合、ネットワークプロファイルが48ページの「NCPの説明」で説明するNCPに相当します。

ネットワークプロファイルでは、いつでも有効または無効にすることができるネットワークインタフェースを指定します。ネットワークプロファイルを使用すると、複数のネットワークインタフェースが使用可能な状況で役立ちます。たとえば、最新のノートパソコンブランドには、有線と無線の両方のインタフェースが搭
ネットワークプロファイルの管理

ネットワークプロファイルの管理

載されています。物理的な場所や作業環境によっては、セキュリティーまたはその他の理由で、一方のインタフェースのみを使用して、他方のインタフェースは無効にする場合があります。

NWAM GUIでは、デフォルトの「自動」ネットワークプロファイアルとユーザー定義のネットワークプロファイルの2種類のネットワークプロファイルが使用可能です。どちらの種類のプロファイルも有効または無効にすることができ、ユーザー定義のプロファイルは変更できますが、「自動」プロファイルはできません。「自動」プロファイルは、NWAM GUIまたはCLIを使用しても作成したり、削除したりできません。ただし、ユーザー定義のプロファイルは、GUIとCLIのどちらかを使用して作成、変更、および削除することができます。

デフォルトでは、「自動」ネットワークプロファイルは最初に1つの有線接続の有効化を試みます。この試みに失敗すると、1つの無線接続の有効化が試みられます。

「ネットワーク設定」ダイアログについて

「ネットワーク設定」ダイアログは、個々のネットワーク接続が構成される場所であり、各ネットワーク接続の現在の状態が表示されます。ダイアログの一番上にあるドロップダウンリストを使用して、切り替え可能なさまざまなビューにアクセスできます。

ダイアログは、次のような方法で開くことができます。

- デスクトップ上の「ネットワークステータス」通知アイコンをクリックします。
- デスクトップパネルのメインメニューから、「システム」→「管理」→「ネットワーク」の順に選択します。
- 「ネットワークステータス」通知アイコンメニューから「ネットワーク設定」を選択します。

「ネットワーク設定」ダイアログの一番上には、「表示」というラベルが付いたドロップダウンリストがあります。このリストでは、各ネットワークプロファイルのネットワーク接続ごとに、「接続ステータス」ビュー、「ネットワークプロファイル」ビュー、および「接続プロパティー」ビューを切り替えることができます。

「接続ステータス」ビュー

「接続ステータス」ビューには、手動起動型のネットワークプロファイル内にある有効な各ネットワーク接続に関する情報、およびアクティブな優先グループ内の各接続（有効か無効かに関係なく）に関する情報が表示されます。「有効な接続」セクションには、有効なすべての接続が、「ネットワークプロファイル」ビューに一覧表示されるときに同じ順序で一覧表示されます。133ページの「有効なネットワーク接続に関する詳細を表示する方法」を参照してください。
「ネットワークプロファイル」ビュー

- ネットワークプロファイルの情報は、「ネットワーク設定」ダイアログの「ネットワークプロファイル」ビューで表示できます。
  このビューを表示するには、「ネットワーク設定」ダイアログの一番上にあるドロップダウンリストで「ネットワークプロファイル」を選択します。

「接続プロパティ」ビュー

- 「接続プロパティ」ビューでは、指定されたネットワーク接続のプロパティを表示および変更できます。このビューに切り替えるには、「表示」ドロップダウンリストから接続名を選択するか、「接続ステータス」と「ネットワークプロファイル」のどちらかのビューで接続名をダブルクリックします。接続のプロパティを表示または編集できるタブ付きのビューが表示されます。
  「接続プロパティ」ビューには、「IPアドレス」タブと「無線」タブの2つのタブが付いています。「無線」タブは、接続タイプが無線の場合にのみ表示されます。この「IPアドレス」タブでは、IPv4とIPv6の両方のアドレスを構成できます。「無線」タブでは、お気に入りネットワークのリストを構成し、使用可能なネットワークに無線インタフェースを接続する方法を選択できます。
ネットワークプロファイルに関する情報を表示する

ネットワークプロファイルの情報は、「ネットワーク設定」ダイアログの「ネットワークプロファイル」ビューで表示できます。

このビューを表示するには、「ネットワーク設定」ダイアログの一番上にあるドロップダウンリストで「ネットワークプロファイル」を選択します。

「ネットワークプロファイル」リストには、使用可能な各ネットワークの名前が表示されます。現在有効になっているプロファイルは、ラジオボタンインジケータ付きで表示されます。デフォルトでは、1つのプロファイル「自動」が表示されます。このプロファイルのアクティブ化はできますが、編集や削除はできません。ただし、複数の追加ネットワークプロファイアルを作成することは可能です。手動で作成されたネットワークプロファイルは、必要に応じてアクティブ化、編集、または削除することができます。

「ネットワークプロファイル」リストの下には、選択したプロファイルの要約が表示されます。選択したプロファイルを完全に表示したり、プロファイルを編集したりするには、「編集」ボタンをクリックします。

注-選択したプロファイルが、有効なプロファイルと異なる場合があります。

あるネットワークプロファイルから別のネットワークプロファイルに切り替える

1. 「ネットワーク設定」ダイアログの「ネットワークプロファイル」ビューを開きます。
ネットワークプロファイルの管理

2. アクティブにするネットワークプロファイルの横にあるラジオボタンを選択します。

3. ネットワークプロファイルを切り替えるには、「了解」をクリックします。プロファイルを切り替えずにダイアログを閉じるには、「取消し」をクリックします。

ネットワークプロファイルを追加または削除する

ネットワークプロファイルを作成または編集するには、「ネットワーク設定」ダイアログの一番上にあるドロップダウンリストから「ネットワークプロファイル」を選択します。

- 新しいネットワークプロファイルを作成するには、「追加」ボタンをクリックしてから、新しいプロファイルの名前を入力します。
- 既存のネットワークプロファイルを複製するには、リスト内でプロファイルを選択し、「複製」ボタンをクリックしてから、新しいプロファイルの名前を入力します。
- ネットワークプロファイルを削除するには、リスト内でプロファイルを選択してから、「削除」ボタンをクリックします。

注 - 「自動」ネットワークプロファイルを削除することはできません。

追加または複製したプロファイルの編集の詳細は、141ページの「ネットワークプロファイルの編集」を参照してください。

ネットワークプロファイルの編集

新しいネットワークプロファイルを手動で追加するとき、または既存のネットワークプロファイルを複製するときは、新しいプロファイルで有効および無効になるネットワーク接続を指定するように新しいプロファイルを編集する必要があります。

注 - 手動で作成したネットワークプロファイルは、編集および削除することができません。ただし、「自動」ネットワークプロファイルは編集や削除ができません。
ネットワークプロファイルの管理

▼ 「ネットワークプロファイル」ダイアログを開く方法

ネットワークプロファイルを編集するには、「ネットワーク設定」ダイアログの「ネットワークプロファイル」ビューでプロファイルを選択してから、「編集」ボタンをクリックします。

ネットワークプロファイルのリストは、上位2つのグループの最小限の説明で構成されています。たとえば、上記の図で表示した「自動」プロファイルには、次の節で詳細に説明する4つのグループの説明が含まれています。

注 - 「自動」ネットワークプロファイルは、変更や削除ができません。「自動」ネットワークプロファイルが「ネットワークプロファイルの編集」ダイアログで選択されると、すべてのプロファイル編集のボタンおよびドロップダウンリストが常に無効になります。

詳細は、オンラインヘルプを参照してください。

優先グループを操作する

「常に有効」グループ内のネットワーク接続は、選択されたネットワークプロファイルがアクティブなときに常に有効になります。
ネットワーク接続を「常に有効」グループに移動するには、まず接続を選択してから、次のいずれかを行います。

- 「有効」ボタンをクリックします。
- 接続が「常に有効」グループに移動するまで「上へ」ボタンをクリックします。

「常に無効」グループ内のネットワーク接続は、選択されたネットワークプロファイルがアクティブなときに常に無効になります。

ネットワーク接続を「常に無効」グループに移動するには、まず接続を選択してから、次のいずれかを行います。

- 「無効」ボタンをクリックします。
- 接続が「常に無効」グループに移動するまで「下へ」ボタンをクリックします。

1つ以上のネットワークインターフェースを1つのグループとして処理するネットワークプロファイルを作成できます。優先度が最高のグループで1つ以上のインタフェースを有効にすることができない場合は、グループの優先度タイプに従って、次に優先度が高いグループが考慮されます。

次の表は、使用可能な3つの異なる優先グループを示しています。

<table>
<thead>
<tr>
<th>優先度タイプ</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exclusive</td>
<td>グループ内の1つの接続が有効になり、その他のすべての接続は無効になります。グループ内の1つ以上の接続が有効である限り（必ずしも常に同じ接続とは限りません）、優先度が低いグループ内の接続の有効化は試みられません。</td>
</tr>
<tr>
<td>共有</td>
<td>有効化可能なグループ内のすべての接続が有効になります。グループ内の1つ以上の接続が有効のままである限り、優先度が低いグループ内の接続の有効化は試みられません。</td>
</tr>
<tr>
<td>All</td>
<td>グループ内のすべての接続が有効になります。接続のいずれかが失われると、グループ内のすべての接続が無効になります。グループ内のすべての接続が有効のままである限り、優先度が低いグループ内の接続の有効化は試みられません。</td>
</tr>
</tbody>
</table>

たとえば、デフォルトの「自動」ネットワークプロファイルには、排他的な2つの優先グループが含まれています。優先度が高い方のグループには、すべての有線ネットワーク接続が含まれています。優先度が低い方のグループには、すべての無線ネットワーク接続が含まれています。

このようなタスクまたはその他のタスクの実行の詳細な手順については、オンラインヘルプを参照してください。
ネットワークプロファイルの管理

場所を作成および管理する

場所は、ネームサービスやファイアウォールの設定などのネットワーク構成の特定要素で構成されています（必要に応じて、同時に適用されます）。さまざまな使用目的で複数の場所を作成できます。たとえば、オフィスで会社のインタラネットを使用して接続しているときは、1つの場所を使用できます。無線アクセスポイントを使用してパブリックインターネットに接続しているときは、自宅で別の場所を使用できます。場所は、ネットワーク接続で取得されるIPアドレスなどの環境条件に従って手動で、または自動的にアクティブにできます。

「ネットワークの場所」ダイアログでは、場所を切り替えたり、場所のプロパティを編集したり、新しい場所を作成したり、場所を削除したりできます。作成および削除できるのはユーザー定義の場所だけであることに注意してください。「場所」ダイアログは、「ネットワーク設定」ダイアログの「接続ステータス」ビューから開くことができます。

「場所」リストは、「ネットワークステータス」通知のアイコンメニューのリストと似ています。使用可能な各場所は、起動型を表すアイコン付きで一覧表示されます。

場所のタイプは次のとおりです。

- システム - このタイプの場所は、システム定義の場所（「自動」および「ネットワークなし」）です。つまり、現在のネットワーク状況に基づいて、場所をアクティブにするタイミングがシステムにより決定されます。
- 手動 - このタイプの場所は、「ネットワークの場所」ダイアログを使用するか、「ネットワークステータス」通知アイコンと対話することによって、手動で有効または無効することができます。
ネットワークプロファイルの管理

- 条件付き - このタイプの場所は、場所の作成中に指定される規則に従って、自動的に有効または無効になります。

「選択された場所」ドロップダウンリストには、選択された起動型も表示されます。有効になっている場所は、リストの一番目の列に表示されるラジオボタンの選択で示されます。

▼ 場所の起動モードを変更する方法
次のタスクでは、NWAM GUIを使用して場所の起動モードを変更する方法について説明します。netcfgコマンドを使用する場合は、指定された場所のプロバティを変更して起動モードを変更します。詳細は、100ページの「プロファイルのプロバディー値を設定および変更する」を参照してください。

1 「ネットワークステータス」通知アイコンの「場所」サブメニューから、「ネットワークの場所」を選択します。または、「ネットワーク設定」ダイアログの「接続ステータス」ビューから、「場所」ボタンをクリックします。

2 場所の起動モードを変更するには、リスト内で場所を選択してから、「選択された場所」ドロップダウンリストから新しい起動モードを選択します。

注 - システムの場所が選択されているときは、ドロップダウンリストに「システムによって起動されました」が表示され、ドロップダウンリストと「規則の編集」ボタンの両方が無効になることに注意してください。

- 「手動」または「条件付き」の場所が選択されているときは、ドロップダウンリストのオプションは次のとおりです。

- 手動起動のみ: この場所は、手動で選択した場合にのみ有効になります。このオプションが選択されると、「規則の編集」ボタンが無効になります。

- 規則によって起動: この場所は、特定のネットワーク条件で自動的に選択されます。このオプションが選択されると、「規則の編集」ボタンが有効になります。

3 (省略可能) 場所をアクティブにする方法とタイミングに対する規則を設定するには、「規則の編集」ボタンをクリックします。
詳細な手順については、オンラインヘルプの「「規則」ダイアログの操作」を参照してください。

▼ ある場所から別の場所に切り替える方法
次のタスクでは、NWAM GUIを使用してある場所から別の場所に切り替える方法について説明します。CLIを使用して場所を切り替えるには、netadmコマンドを使用して新しい場所をアクティブにします。システムでは常に1つの場所だけをアクティブにする必要があるため、新しい場所をアクティブにすると、暗黙的に現在有
ネットワークプロファイルの管理

効になっている場所は無効になります。ネットワークプロファイルをアクティブにするときにも、同じ規則が適用されます。場所の有効化および無効化の詳細は、118ページの「プロファイルをアクティブまたは非アクティブにする」を参照してください。

● 「ネットワークステータス」通知アイコンの「場所」サブメニューから、アクティブにする場所を選択します。
「場所」サブメニューで「場所の自動切り替え」オプションが選択されている場合は、アクティブにする場所を手動で選択できません。ネットワーク環境の変化に応じて、常に最適な「システム」または「条件付き」の場所が自動的にアクティブになります。
「場所」サブメニューで「場所の手動切り替え」オプションが選択されている場合は、その起動型には関係なく、使用可能な場所をアクティブにできます。選択された場所は、無期限にアクティブなままになります。

あるいは、「ネットワークの場所」ダイアログでも場所を切り替えることができます。このためには、次の手順に従います。

a. 「ネットワークステータス」通知アイコンの「場所」サブメニューから、「ネットワークの場所」を選択します。または、「ネットワーク設定」ダイアログの「接続ステータス」ビューから、「場所」ボタンをクリックします。

b. 切り替える場所のラジオボタンを選択してから、「了解」をクリックします。

- 「ネットワークの場所」ダイアログで「場所の自動切り替え」ラジオボタンが選択されている場合は、アクティブにする場所を手動で選択できません。ネットワーク環境の変化に応じて、常に最適な「システム」または「条件付き」の場所が自動的にアクティブになります。
- 「ネットワークの場所」ダイアログで「場所の手動切り替え」ラジオボタンが選択されている場合は、その起動型には関係なく、使用可能な場所をアクティブにできます。その場所は無期限にアクティブなままになることに注意してください。

場所の編集

NWAM GUIを使用して場所を編集することは、NWAM CLIを使用する場合の場所のプロバティーを変更することに相当します。
場所を編集するには、「ネットワークステータス」通知アイコンの「場所」サブメニューから「ネットワークの場所」を選択します。または、「ネットワーク設定」ダイアログの「接続ステータス」ビューから、「場所」ボタンをクリックします。

指定された場所のプロパティを編集するには、リスト内で場所を選択してから、「編集」をクリックします。

あるいは、リスト内の場所をダブルクリックすることもできます。

「場所の編集」ダイアログが開き、次の2つのタブが使用できます。

ネームサービス：指定された場所内にネームサービスを構成できます。

セキュリティー：指定された場所が有効なときに、IP フィルタや IPsec 機能で使用される構成ファイルを選択できます。

編集する情報を表示するには、該当するタブを選択します。

外部ネットワーク識別子について

外部ネットワーク識別子 (ENM) とは、NWAM の外部にあるアプリケーション用に作成されるプロファイルです。ただし、これらのアプリケーションではネットワーク構成の作成および変更が可能です。たとえば、VPN アプリケーションを使用すると、ネットワーク接続が仮想プライベートネットワークと通信できます。ENM は、NWAM GUI で「ネットワーク修飾子」ダイアログを使用することで構成および監視されます。

注：NWAM GUI を使用してネットワーク修飾子アプリケーションまたはサービスを管理する前に、手動でインストールしてから、証明書や共有シークレットのインストールなどの初期設定を完了する必要があります。

ENM は、必要に応じて手動で起動および停止できます。また、ユーザー定義の規則に従って、自動的に起動することもできます。このダイアログを使用して管理するには、ネットワーク修飾子アプリケーションをコマンド行ツールまたは SMF サービスとして実装する必要があります。

NWAM CLI を使用した ENM の作成および管理の詳細は、93 ページの「ENM プロファイルを作成する」を参照してください。
「ネットワーク修飾子」ダイアログについて

このダイアログは、ネットワーク構成を作成および変更できる外部ネットワーク識別子 (ENM) アプリケーションの追加や削除、起動と停止、および編集を行う際に使用されます。

次のいずれかの方法を使用してダイアログを開きます。

- 「ネットワーク設定」ダイアログの「接続ステータス」ビューで「修飾子」ボタンをクリックします。
- 「ネットワークステータス」通知アイコンを右クリックしてから、「ネットワーク修飾子の設定」メニューユイを選択します。

このダイアログのメインセクションは、各 ENM の次の情報を表示する 3 つの列から成るリストです。

- アクティブ化の状態 (「手動」または「条件付き」)
- ユーザー定義の名前 (たとえば、「Cisco VPN」)
- 現在のステータス (「実行中」または「停止しました」)
「規則に従って開始/停止する」チェックボックスは、選択したネットワーク修飾子アプリケーションの起動型が「条件付き」の場合はチェックされ、起動型が「手動」の場合はチェックが解除されます。起動型を変更するには、チェックボックスを切り替えます。

▼ コマンド行の ENM を追加する方法

次の手順では、コマンド行の ENM を追加する方法について説明します。ネットワーク修飾子アプリケーションサービスの追加の詳細は、オンラインヘルプを参照してください。

1 次のいずれかの方法を使用して「ネットワーク修飾子」ダイアログを開きます。
   - 次に、「ネットワーク設定」ダイアログの「接続ステータス」ビューから、「修飾子」ボタンをクリックします。
   - 「ネットワークステータス」通知アイコンを右クリックしてから、「ネットワーク修飾子の設定」メニュー項目を選択します。

2 「追加」ボタンをクリックします。

3 新しいネットワーク修飾子アプリケーションの名前を入力します。

4 次のいずれかを行います。
   - 起動型が「手動」である新しいエントリを追加するには、Enter または Tab キーを押します。
     2つの修飾子の管理のラジオボタンを有効にします。1 つ目の「コマンド行アプリケーション」は、デフォルトで選択されています。「開始コマンド」および「停止コマンド」フィールドと、2つの「参照」ボタンも有効になっています。
   - 変更を取り消すには、Esc キーを押します。

5 「開始コマンド」フィールドに、このネットワーク修飾子アプリケーションを開始するコマンドを入力します。
   あるいは、「参照」ボタンを使用して、使用するコマンドを選択できるファイルチューダダイアログを開くこともできます。
   このネットワーク修飾子アプリケーションの「開始」ボタンは、このフィールドに有効なコマンドが入力されるまで無効になっています。
外部ネットワーク識別子について

6 「停止コマンド」フィールドに、このネットワーク修飾子アプリケーションを停止するコマンドを入力します。
あるいは、「参照」ボタンを使用して、使用するコマンドを選択できるファイルチャーザダイアログを開くこともできます。
このネットワーク修飾子アプリケーションの「停止」ボタンは、このフィールドに有効なコマンドが入力されるまで無効になっています。

7 このアプリケーションを追加するには、「了解」をクリックします。
外部ネットワーク修飾子が追加されます。
パート Ⅱ

データリンクとインタフェース構成

ここでは、パートⅠ「Network Auto-Magic」で導入されたネットワーク構成プロファイルのコンテキストでのデータリンクとインタフェース構成の手順について説明します。これらの手順は、有効またはアクティブになっている任意の固定プロファイルに適用されます。
プロファイルでのデータリンクおよびインタフェース構成コマンドの使用

この章は、dladmやipadmのような従来の構成コマンドがプロファイルベースのネットワーク構成と関係する場合の、これらのコマンドの使用方法について説明します。

プロファイルベースのネットワーク構成の特徴

このOracle Solarisリリースでは、ネットワーク構成はプロファイルに基づいています。システムのネットワーク構成の設定は、特定のネットワーク構成プロファイル (NCP) と対応する場所プロファイルによって管理されます。NCP、場所プロファイル、およびその他のプロファイルタイプ、それらのプロバティー、およびプロファイルの操作や管理に使用するコマンドの詳細については、パートI「Network Auto-Magic」を参照してください。

注 - ネットワーク構成に関する主なプロファイルタイプは、NCP、場所プロファイル、外部ネットワーク修飾子 (ENM)、および無線ローカルエリアネットワーク (WLAN) です。これらのタイプのうち、主要なプロファイルはNCPです。このマニュアル全体を通じて、特に明記しないかぎり、「プロファイル」という用語はNCPを指します。

プロファイルベースのネットワーク構成の特徴は、次のとおりです。

- あるシステムのネットワーク構成を管理するために一度にアクティブになるのは、1組のNCPと場所プロファイルだけです。システム内のほかの既存のNCPはすべて、非運用状態になります。
- アクティブなNCPは、「リアクティブ」、「固定」のいずれかになります。リアクティブプロファイルでは、システムのネットワーク環境の変化に適応できるように、ネットワーク構成が監視されます。固定プロファイルでは、ネットワーク構成はインスタンス化されますが、監視されません。
プロファイルと構成ツール

- NCP のさまざまなプロバティーの値が、そのプロファイルによるネットワーク構成の管理方法を制御するポリシーを構成します。
- NCP のプロバティーに対する変更は、新しいプロバティー値として即座に実装され、そのプロファイルの、ネットワーク構成を管理するポリシーの一部となります。

注 - Oracle Solaris 1111 Express リリースからアップグレードされたシステムでは、アップグレード前に使用されていたネットワーク構成が、アップグレード後のアクティブプロファイルとなります。以前の構成がdladm およびipadm コマンドによって作成されていた場合、その構成はプロファイル DefaultFixed となり、これがシステムでアクティブになります。それ以外の場合、その構成はプロファイル Automatic となり、これがシステムのネットワーク構成を管理します。

プロファイルと構成ツール

プロファイルのカスタマイズに使用するツールは、アクティブなプロファイルに依存します。アクティブなプロファイルが Automatic などのリアクティブプロファイルである場合は、netcfg およびnetadm コマンドを使用してプロファイルを構成および監視します。アクティブなプロファイルがDefaultFixed などの固定プロファイルである場合は、dladm およびipadm コマンドを使用します。

dladm およびipadm コマンドはアクティブなプロファイルに対してのみ有効です。したがって、これらのコマンドを使用する前に、次を確認する必要があります。

- 適切なコマンドを使用して正しいターゲットプロファイルに対して変更を行うよう、どのプロファイルがアクティブであるかを知る。
- コマンド使用後に予期しない構成動作が発生しないように、ターゲットプロファイルがリアクティブ、固定のいずれであるかを知る。リアクティブプロファイルは、固定プロファイルと異なる方法でネットワーク構成を管理します。したがって、変更実装時の 2 つのプロファイルの動作も異なります。

注 - dladm およびipadm コマンドの. tオプションを使用して一時的な設定を作成できるのは、固定プロファイルの場合だけです。リアクティブプロファイルでは、このオプションはサポートされません。

dladm およびipadm コマンドをプロファイル上で正しく使用するには、次の 2 つの手順に従います。
ネットワーク管理モードを判定する方法

Automaticなどのリアクティブ NCPがシステムでアクティブなNCPである場合、システムのネットワーク管理モードは自動です。ネットワーク構成を実行する前に、次の手順に従ってネットワーク管理モードを判定してください。この手順により、正しいコマンドを使用して、適切なプロファイル上で構成を実装していることを確認できます。

1 システム内のプロファイルを一覧表示します。

```bash
# netadm list -x
```

<table>
<thead>
<tr>
<th>TYPE</th>
<th>PROFILE</th>
<th>STATE</th>
<th>AUXILIARY STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ncp</td>
<td>Automatic</td>
<td>online active</td>
<td></td>
</tr>
<tr>
<td>ncu:phys</td>
<td>net0</td>
<td>online</td>
<td>interface/link is up</td>
</tr>
<tr>
<td>ncu:ip</td>
<td>net0</td>
<td>online</td>
<td>interface/link is up</td>
</tr>
<tr>
<td>ncu:phys</td>
<td>net1</td>
<td>online</td>
<td>interface/link is up</td>
</tr>
<tr>
<td>ncp</td>
<td>testcfg</td>
<td>disabled</td>
<td>disabled by administrator</td>
</tr>
<tr>
<td>loc</td>
<td>Automatic</td>
<td>offline*</td>
<td>waiting for IP address to be set</td>
</tr>
<tr>
<td>loc</td>
<td>NoNet</td>
<td>offline</td>
<td>conditions for activation are unmet</td>
</tr>
<tr>
<td>loc</td>
<td>Lab</td>
<td>online</td>
<td>active</td>
</tr>
<tr>
<td>loc</td>
<td>User</td>
<td>disabled</td>
<td>disabled by administrator</td>
</tr>
</tbody>
</table>

出力には、次の2種類の情報が示されます。

- netadm list コマンドは、ネットワーク管理モードが自動の場合にのみサポートされます。このため、プロファイルリストが生成されるということは、ネットワーク管理が自動モードであることを示します。そうでない場合、netadm list コマンドは次のメッセージを生成して、代わりに DefaultFixed プロファイルがアクティブであることを示します。

  netadm: DefaultFixed NCP is enabled; automatic network management is not available. 'netadm list' is only supported when automatic network management is active.

- プロファイルリストが生成される場合には、リアクティブ NCP の online 状態により、どの NCP が有効であるかも示されます。サンプル出力は、既存のリアクティブ NCP は Automatic NCP のみであることを示しています。ユーザーが作成したほかの NCP がシステムに存在する場合には、それらもリストに含まれます。

2 使用する構成ツールに対して、適切なプロファイルがアクティブになっていることを確認してください。

たとえば、dladmおよびipadmコマンドは、DefaultFixed プロファイルでのみ使用できます。ただし、netcfg コマンドを使用できるのは、ネットワーク管理が自動モードである Automatic などのリアクティブなプロファイルに対してだけです。

選択した構成ツールで変更するプロファイルを含むプロファイルがアクティブでない場合は、次の手順に進んで、適切なプロファイルを有効にしてください。該当するプロファイルがアクティブな場合は、ツールの使用を開始して、ネットワーク構成できます。

たとえば、ネットワーク管理を自動モードにせずに、dladmやipadmなどのコマンド行を使ってデータリンクおよびインタフェースを手動で構成する場合を考えま
3 別のプロファイルを構成する場合は、次のように入力してそのプロファイルを有効にします。

```
# netadm enable -p ncp profile-name
例:
```

```
# netadm enable -p ncp defaultfixed
ネットワーク管理が自動モードで、別のリアクティブ NCP を使用する場合にも、同じコマンド構文を使用します。手順 1 のサンプル出力に基づき、Automatic の代わりに、ユーザーアクティブの NCP testcfg をアクティブにするとしますよ。その場合、次のように入力します。

```

```
# netadm enable -p ncp testcfg
```

注意- このコマンドはアクティブなプロファイルを切り替えます。アクティブなプロファイルを切り替えると、既存のネットワーク構成は削除され、新しい構成が作成されます。以前アクティブだった NCP 上で実装された永続的な変更はすべて、新しいアクティブな NCP では除外されます。

次の手順

次の各章は、さまざまな種類のデータリンク構成やインタフェース構成を実行するために使用可能な手順について説明しています。

- データリンクを構成する場合は、第 8 章「データリンクの構成と管理」を参照してください。
- IP インタフェースを構成する場合は、第 9 章「IP インタフェースの構成」を参照してください。
- ワイヤレスインタフェースを構成する場合は、第 10 章「Oracle Solaris 上での無線インタフェース通信の構成」を参照してください。
- ブリッジを構成する場合は、第 11 章「ブリッジを管理する」を参照してください。
- リンク集約を構成する場合は、第 12 章「リンク集約の管理」を参照してください。
- VLAN を構成する場合は、第 13 章「VLAN の管理」を参照してください。
- IPMP グループを構成する場合は、第 14 章「IPMP の紹介」と第 15 章「IPMP の管理」を参照してください。
- リンク層検出プロトコル (LLDP) を構成する場合は、第 16 章「LLDP によるネットワーク接続情報の交換」を参照してください。
データリンクの構成と管理

この章は、dladmコマンドと、このコマンドを使用してデータリンクを構成する方法について説明します。

データリンクの構成(タスク)

次の表は、dladmコマンドを使用して実行可能なさまざまなデータリンク構成タスクの一覧を示します。またこの表は、タスクを完了するための手順へのリンクも示します。

表8-1 基本的なデータリンク構成の実行(タスクマップ)

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>データリンクの名前を変更します。</td>
<td>ハードウェアベースの名前を使用しないでデータリンク名をカスタマイズします。</td>
<td>160ページの「データリンクの名前を変更する方法」</td>
</tr>
<tr>
<td>データリンクの物理属性を表示します。</td>
<td>メディアのタイプや関連するデバイスインスタンスの情報など、データリンクのベースとなる物理情報の一覧表示します。</td>
<td>161ページの「データリンクの物理属性に関する情報を表示する方法」</td>
</tr>
<tr>
<td>データリンクの状態を表示します。</td>
<td>データリンクのステータスに関する情報を一覧表示します。</td>
<td>162ページの「データリンクの情報を表示する方法」</td>
</tr>
<tr>
<td>データリンクを削除します。</td>
<td>もう使用されていないNICに関連付けられたリンク構成を削除します。</td>
<td>163ページの「データリンクを削除する方法」</td>
</tr>
</tbody>
</table>
dladm コマンド

dladm コマンドは、GLDv3 ドライバ構成フレームワークを完全に実装したあと、時間の経過とともに拡張機能を獲得してきました。このフレームワークは次のように、NIC ドライバの構成を拡張します。

- ネットワークドライバのプロバティーの構成に必要となるのは、単一のコマンドインタフェースである dladm コマンドだけです。
- プロバティーにかかわらず、dladm subcommand properties datalink という一貫した構文が使用されます。
- dladm コマンドの使用は、ドライバの公開プロバティーと非公開プロバティーの両方に適用されます。
dladmコマンド

- 特定のドライバに対してdladmコマンドを使用しても、似たタイプのほかのNICのネットワーク接続に支障が生じることはありません。したがって、データリンクのプロパティーを動的に構成できます。
- データリンク構成の設定はdladmリポジトリ内に格納され、システムのリブート後も永続します。

データリンクの構成時に前述のメリットの恩恵を受けるには、nddコマンドのような以前のリリースの従来型ツールではなく、dladmを構成ツールとして使用するようにしてください。

データリンクを管理するには、次のdladmサブコマンドを使用します。
- dladm rename-link は、データリンクの名前を変更します。
- dladm show-link は、システム内の既存のデータリンクを表示します。
- dladm show-phys は、データリンクの物理属性を表示します。
- dladm delete-phys は、データリンクを削除します。
- dladm show-linkprop は、データリンクに関連付けられたプロパティーを表示します。
- dladm set-linkprop は、指定されたデータリンクプロパティーを設定します。
- dladm reset-linkprop は、プロパティーをデフォルト設定に戻します。
- dladm show-ether は、データリンクのEthernetパラメータ設定を表示します。

dladmコマンドは、次のようなほかの種類のリンク管理を実行する場合にも使用されます。
- プリッジの構成。第11章「プリッジを管理する」を参照
- リンク集約の構成。第12章「リンク集約の管理」を参照
- VLANの構成。第13章「VLANの管理」を参照
- トンネルの構成。『Oracle Solarisの管理: IP サービス』の第6章「IP トンネルの構成」を参照。

コマンドの詳細については、dladm(1M)のマニュアルページを参照してください。

次の場合、dladmコマンドを使用してデータリンクを構成する方法を示します。ほとんどの場合、データリンクの構成は、そのリンク上でのIPインタフェースの構成の一部となります。したがって、該当する場合は、これらの手順に、ipadmコマンドを使用したIPインタフェースの構成手順を含めています。ただし、IPインタフェースの構成とipadmコマンドについては、第9章「IPインタフェースの構成」で詳しく説明します。
▼ データリンクの名前を変更する方法

この手順は、データリンクの名前をカスタマイズされた名前に変更する場合に使用します。たとえば、アップグレードされたシステムのいくつかのデータリンクでハードウェアベースのレガシー名が維持されていて、それらの名前を汎用名に変更するような場合です。

始める前に リンク名の変更の影響を受ける可能性のある関連構成について実行する必要があるその他の手順の調査と準備が完了していることを確認します。詳細は、29ページの「アップグレードされたシステムでのリンク名」を参照してください。

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 データリンク上にIPインタフェースが構成されている場合は、そのIPインタフェースを削除します。

   # ipadm delete-ip interface

3 リンクの現在のリンク名を変更します。

   # dladm rename-link old-linkname new-linkname

   old-linkname データリンクの現在の名前を表します。リンク名はデフォルトで、bge0のようなハードウェアベースのものになります。

   new-linkname データリンクに割り当てる任意の名前を表します。リンク名を割り当てる場合の規則については、30ページの「有効なリンク名の規則」を参照してください。29ページの「アップグレードされたシステムでのリンク名」も参照し、データリンク名の変更に関する詳細情報を確認してください。

システムのリブート後に新しいリンク名が反映しないようにするには、サブコマンドの直接に-tオプションを使用します。このオプションは、リンク名を一時的に変更します。システムをリブートすると、元のリンク名に戻ります。

注 - dladm rename-link を使用すると、リンクの構成であるデータリンクから別のデータリンクに転送できます。例については、173ページの「動的再構成を使用してネットワークインタフェースカードを交換する方法」を参照してください。リンク名をこの目的で変更する場合には、構成を継承するリンクに既存の構成が一切含まれていないことを確認してください。それ以外の場合、転送が失敗します。
例8-1 システムの主ネットワークインタフェースの変更
次の例は、データリンクの名前を変更することでシステムの主ネットワークインタフェースを2番目のNICに切り替える方法を示します。システムの主ネットワークインタフェースはnet0ですが、これは、e1000g0上のデータリンクの汎用名です。この主ネットワークインタフェースのベースとなるインタフェースを、e1000g0からnge0に切り替えます。この例は、新しいブート環境を作成する手順の一部として使用できます。

```
# dladm show-phys
<table>
<thead>
<tr>
<th>LINK</th>
<th>MEDIA</th>
<th>STATE</th>
<th>SPEED</th>
<th>DUPLEX</th>
<th>DEVICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>net0</td>
<td>Ethernet</td>
<td>up</td>
<td>1000</td>
<td>full</td>
<td>e1000g0</td>
</tr>
<tr>
<td>net1</td>
<td>Ethernet</td>
<td>up</td>
<td>1000</td>
<td>full</td>
<td>nge0</td>
</tr>
</tbody>
</table>

# dladm rename-link net0 oldnet0
# dladm rename-link net1 net0

# dladm show-phys
<table>
<thead>
<tr>
<th>LINK</th>
<th>MEDIA</th>
<th>STATE</th>
<th>SPEED</th>
<th>DUPLEX</th>
<th>DEVICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>oldnet0</td>
<td>Ethernet</td>
<td>up</td>
<td>1000</td>
<td>full</td>
<td>e1000g0</td>
</tr>
<tr>
<td>net0</td>
<td>Ethernet</td>
<td>up</td>
<td>1000</td>
<td>full</td>
<td>nge0</td>
</tr>
</tbody>
</table>
```

▼ データリンクの物理属性に関する情報を表示する方法
この手順は、システムのデータリンクの物理属性に関する情報を表示する手順を示します。

1 管理者になります。
詳細は、『Oracle Solarisの管理：セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 システム上に現在存在しているデータリンクの物理属性に関する情報を表示します。
```
# dladm show-phys
このコマンドで-Pを使用すると、各リンクのフラグステータスも表示できます。関連ハードウェアが取り外された場合、データリンクは使用不可能になります。-Pオプションを指定しないと、このコマンドは使用可能なデータリンクのみを表示します。
データリンクの/devicesパスを表示するには、-vオプションを使用します。
```
dladmコマンド

例8-2 使用可能なデータリンクの表示

次の例では、-PオプションによってFLAGS列が含められていますが、ここには、使用可能なリンクは示されます。データリンクnet0のrフラグは、このリンクに関連付けられたハードウェア(nge)が取り外されたことを示しています。

```
# dladm show-phys
LINK MEDIA STATE SPEED DUPLEX DEVICE
net0 Ethernet up 100Mb full e100g0
net1 Infiniband down 0Mb -- ibd0
net3 Ethernet up 100Mb full bge0
net4 Ethernet -- 0Mb -- nge0
```

次の例は、-Lオプション使用時に表示されるリンクとその物理的な場所を示しています。

```
# dladm show-phys -L
LINK DEVICE LOCATION
net0 bge0 MB
net2 ibp0 MB/RISER0/PCIE0/PORT1
net3 ibp1 MB/RISER0/PCIE0/PORT2
net4 eoib2 MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

▼ データリンクの情報を表示する方法

この手順は、使用可能なリンクのステータスを表示します。

1 管理者になります。
   詳細は、『Oracle Solarisの管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 リンクの情報を表示します。
   # dladm show-link

例8-3 使用可能なリンクの表示

次の例は、システム上の永続で使用可能なリンクを表示します。

```
# dladm show-link -P
LINK CLASS BRIDGE OVER
net0 phys -- --
net1 phys -- --
net2 phys -- --
```
dladm コマンド

-P オプションは、永続的ではあるが使用不可能になっている既存リンクもすべて表示します。永続的なリンクを一時的に削除すると、そのリンクは使用不可能になります。リンクの関連ハードウェアを取り外した場合も、リンクが使用不可能になります。

▼ データリンクを削除する方法

この手順は、NICに関連付けられたリンク構成を削除します。交換する予定のないNICを切り離した場合、そのNICに関連付けられたリンク構成を削除できます。この手順を完了したら、リンク名を再利用できます。

1 管理者になります。詳細は、『Oracle Solaris の管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 ハードウェアが取り外されたリンクも含め、システム上のデータリンクを表示します。取り外されたハードウェアに関する情報を含めるには、-P オプションを使用します。

   # dladm show-phys

3 交換する予定のない取り外されたハードウェアのリンク構成を削除します。

   # dladm delete-phys link

例 8–4 データリンクの削除

次の例で、net2 の r フラグは、このリンクの関連ハードウェア (e1000g0) が取り外されたことを示しています。したがって、リンク net2 も削除し、その名前を新しいデータリンクに割り当て直すことができます。

   # dladm show-phys -P

<table>
<thead>
<tr>
<th>LINK</th>
<th>DEVICE</th>
<th>MEDIA</th>
<th>FLAGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>net0</td>
<td>nge0</td>
<td>Ethernet</td>
<td>-----</td>
</tr>
<tr>
<td>net1</td>
<td>bge0</td>
<td>Ethernet</td>
<td>-----</td>
</tr>
<tr>
<td>net2</td>
<td>e1000g0</td>
<td>Ethernet</td>
<td>r----</td>
</tr>
</tbody>
</table>

   # dladm delete-phys net2
データリンクプロパティーの設定

dladmコマンドは、データリンクの基本的な構成を実行するためだけでなく、データリンクプロパティーを設定したり、ネットワークの要求に従ってそれらのプロパティーをカスタマイズしたりするために使用することもできます。

注-dladmコマンドを使用してデータリンクプロパティーをカスタマイズできるのは、e1000というように、リングのネットワークドライバがGLDv3フレームワーク用に変換されている場合です。特定のドライバがこの機能をサポートしているかどうか確認するには、ドライバのマニュアルページを参照してください。

データリンクプロパティーの概要

カスタマイズ可能なデータリンクプロパティーは、ある特定のNICドライバがサポートするプロパティーに依存します。dladmコマンドを使用して構成可能なデータリンクプロパティーは、2つのカテゴリのいずれかに該当します:

- Ethernet用のリンク速度や自動ネゴシエーション、またはすべてのデータリンクドライバで適用可能なMTUサイズなど。特定メディアタイプの任意のドライバに適用可能な「公開プロパティー」。
- 特定メディアタイプのNICドライバの特定のサブセットに固有の「非公開プロパティー」。これらのプロパティーはそのサブセットに固有のものとなる可能性がありますが、それは、これらのプロパティーが、ドライバに関連付けられたハードウェアに密接に関連しているか、あるいはデバッガ関連のチューニング可能パラメータのように、ドライバ実装自体の詳細に密接に関連しているからです。

リンクプロパティーは通常、デフォルト設定を持ちます。ただし、特定のネットワークシナリオでは、データリンクの特定のプロパティー設定を変更する必要がある場合もあります。これらのプロパティー設定は、公開プロパティー、非公開プロパティーのいずれかになります。たとえば、NICは、自動ネゴシエーションを正しく実行しない古いスイッチと通信する可能性があります。また、スイッチは、ジャンポフレームをサポートするように構成されている可能性があります。あるいは、特定のドライバで、パケットの送信やパケットの受信を制御するドライバ固有のプロパティーを変更する必要がある可能性もあります。Oracle Solarisでは、これらすべての設定を、単一の管理ツールdladmによって設定し直せるようになりました。
dladmコマンドを使用したデータリンクプロパティの設定

次の節は、特定のデータリンクプロパティを設定するための手順を、例とともに提供します。選択されたプロパティは公開プロパティであり、すべての NIC ドライバに共通するものです。別の節は、ドライバ固有のデータリンクプロパティについて説明します。この節のあとに、e1000g ドライバの選択された非公開プロパティを構成するための手順が続きます。

▼ ジャンボフレームのサポートを有効にする方法
ネットワーク設定でのジャンボフレームのサポートの有効化は、大部分のネットワークシナリオで一般的なタスクです。ジャンボフレームをサポートするには、データリンクの最大伝送単位 (MTU) のサイズを増やす必要があります。次の手順には、データリンクを識別するためのカスタマイズされた名前の使用が含まれています。カスタマイズされた名前の概要やそのネットワーク構成での使用方法については、22 ページの「Oracle Solaris のネットワークスタック」を参照してください。

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 MTU サイズを設定し直す必要がある特定の Ethernet デバイスを識別するために、システム内のリンクを表示します。
# dladm show-phys
この手順は特に、ネットワーク構成がデータリンクに対してカスタマイズされた名前を使用している場合に実行します。カスタマイズされた名前を使用する場合、データリンクは必ずしもハードウェアベースの名前で識別されなくなります。たとえば、Ethernet デバイスは bge0 です。しかし、そのデバイス上のデータリンクの名前が net0 に変更されます。したがって、net0 の MTU サイズを構成する必要があります。カスタマイズされた名前を使用したデータリンクに関する構成タスクの例については、181 ページの「IP インタフェースの構成 (タスク)」を参照してください。

3 (省略可能) データリンクの現在の MTU サイズやその他のプロパティを表示します。
- データリンクの特定のプロパティを表示する場合は、次の構文を使用します。
  dladm show-linkprop -p property datalink
  このコマンドは、指定期されたプロパティの設定を表示します。
データリンクプロパティーの設定

- データリンクの選択されたいくつかのプロパティーを表示する場合は、次の構文を使用します。
  
  # dladm show-link datalink
  このコマンドは、MTU サイズなど、データリンクの情報を表示します。

4 データリンク上に IP インタフェースが構成されている場合は、その IP インタフェースを削除します。

  # ipadm delete-ip interface

5 リンクの MTU サイズを、ジャンボフレーム用の設定である 9000 に変更します。

  # dladm set-linkprop -p mtu=9000 datalink

6 IP インタフェースを作成します。

  # ipadm create-ip interface

7 IP インタフェースを構成します。

  # ipadm create-addr -T addr-type [-a address] addrobj
  ipadm コマンドの詳細については、ipadm(1M) を参照してください。

8 (省略可能) インタフェースが新しい MTU サイズを使用していることを、手順 3 のいずれかのコマンドを実行して確認します。

  # dladm show-linkprop -p mtu datalink

9 (省略可能) リンクの現在の Ethernet 設定を表示します。

  # dladm show-ether datalink

例8-5 ジャンボフレームのサポートの有効化

ジャンボフレームのサポートを有効化する次の例は、次のシナリオに基づいています。

- bge0 と bge1 の 2 つの bge NIC が、システムに含まれています。
- デバイス bge0 は主インタフェースとして使用され、デバイス bge1 はテスト目的で使用されます。
- bge1 ではジャンボフレームのサポートを有効にしますが、主インタフェースのデフォルトの MTU サイズは維持します。
- ネットワーク構成は、データリンクに対してカスタマイズされた名前を使用しています。bge0 のリンク名は net0 です。bge1 のリンク名は net1 です。

  # dladm show-phys
  LINK MEDIA STATE SPEED DUPLEX DEVICE
  net0 ether up 100Mb full bge0
  net1 ether up 100Mb full bge1

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
データリンクプロバティーの設定

```
net2 ether up 100Mb full nge3
# dladm show-linkprop -p mtu net1
LINK PROPERTY VALUE DEFAULT POSSIBLE
net1 mtu 1500 1500 --
# ipadm delete-ip net1
# dladm set-linkprop -p mtu=9000 net1
# ipadm create-ip net1
# ipadm create-addr -T static -a 10.10.1.2/35 net1/v4
# dladm show-link web1
LINK CLASS MTU STATE BRIDGE OVER
web1 phys 9000 up -- --
```

MTUの設定が9000になっていることがわかります。この例では、dladmコマンドを使用してnet1のMTUサイズを直接変更できます。nddコマンドを使用する以前の方法では、net0を削除する必要がありましたが、そのために、主インタフェースの動作が不必要に中断されていました。

### リンク速度パラメータを変更する方法

ほとんどのネットワーク設定は、異なる速度機能を備えたシステムの組み合わせから構成されます。たとえば、比較的古いシステムと新しいシステムの間で通知される速度は、通信を可能にするために低い設定に変更する必要がある場合があります。デフォルトでは、NICカードのすべての速度機能と二重機能が通知されます。この手順は、ギガビット機能をオフにしてメガビット機能のみを通知する方法を示します。

1. 管理者になります。
詳細は、「Oracle Solarisの管理: セキュリティサービス」の「管理権限を取得する方法」を参照してください。

2. (省略可能)変更するプロバティーの現在のステータスを表示します。
```
# dladm show-linkprop -p property datalink
```

3. 低い速度機能を通知する場合は、高い速度機能をオフにし、高い速度機能が通知されないようにします。
```
# dladm set-linkprop -p property=value1 datalink
```

例8-6 NICのギガビット機能の通知の無効化

この例は、リンクnet1がギガビット機能を通知しないようにする方法を示します。

```
# dladm show-linkprop -p adv_1000fdx_cap net1
LINK PROPERTY VALUE DEFAULT POSSIBLE
net1 adv_1000fdx_cap 1 -- 1,0
```
データリンクプロパティーの設定

```
# dladm show-linkprop -p adv_1000hdx_cap web1
LINK PROPERTY VALUE DEFAULT POSSIBLE
net1 adv_1000hdx_cap 1 -- 1,0

リンクのギガビット機能を通知するプロパティーは、adv_1000fdx_cap と adv_1000hdx_cap です。これらのプロパティーを無効にして通知されないようにするには、次のコマンドを入力します。

# dladm set-linkprop -p adv_1000fdx_cap=0 net1
# dladm set-linkprop -p adv_1000hdx_cap=0 net1
```

Ethernet パラメータの設定を一覧表示すると、次の出力が表示されます。

```
# dladm show-ether net1
LINK PTYPE STATE AUTO SPEED-DUPLEX PAUSE
net1 current up yes 1G-f both
```

データリンクプロパティーに関するステータス情報を取得する方法

Ethernet パラメータ設定、リンクプロパティーのいずれかを表示することで、データリンクのプロパティーに関する情報を取得できます。

1 管理者になります。
詳細は、「Oracle Solaris の管理: セキュリティサービス」の「管理権限を取得する方法」を参照してください。

2 Ethernet パラメータ設定に関する情報を取得する場合は、次のコマンドを使用します。
```
# dladm show-ether [-x] datalink
```
ここで、-x オプションを指定すると、リンクに関する追加のパラメータ情報が含められます。-x オプションを指定しないと、現在のパラメータ設定のみが表示されます。

3 リンクのすべてのプロパティーに関する情報を取得する場合は、次のコマンドを使用します。
```
# dladm show-linkprop datalink
```

例 8-7 Ethernet パラメータ設定の表示
この例は、指定されたリンクに関するパラメータ情報の拡張リストを表示します。

```
# dladm show-ether -x net1
LINK PTYPE STATE AUTO SPEED-DUPLEX PAUSE
net1 current up yes 1G-f both
-- capable -- yes 1G-fh,100M-fh,10M-fh both
```

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
データリンクプロパティーの設定

- "adv" のオプションを使用すると、このコマンドは指定されたリンクの組み込み機能や、ホストとリンクパートナーとの間で現在通知されている機能も表示します。次の情報が表示されています。

- Ethernet デバイスの現在の状態については、リンクは稼働しており、毎秒1ギガビットの全二重で機能しています。その自動ネゴシエーション機能が有効化されており、ホストとリンクパートナーの両方が一時停止フレームを送受信できる方向フロー制御が備わっています。

- 現在の設定にかかわらず、Ethernet デバイスの機能が表示されています。ネゴシエーションタイプは自動に設定可能であり、このデバイスは、毎秒1ギガビット、毎秒100メガビット、および毎秒10メガビットの速度を、全二重、半二重の両方でサポートできます。同様に、一時停止フレームはホストとリンクパートナーとの間でどちらの方向にも送受信できます。

- net1 の機能が次のように通知されています: 自動ネゴシエーション、速度-二重、一時停止フレームのフロー制御。

- 同様に、net1 のリンクパートナーつまりビアパートナーが、次の機能を通知していいます: 自動ネゴシエーション、速度-二重、一時停止フレームのフロー制御。

例 8-8 リンクプロパティーの表示

この例は、リンクのすべてのプロパティーを一覧表示する方法を示します。特定のプロパティーのみを表示する場合は、監視するその特定のプロパティーで -p オプションを使用します。

```bash
# dladm show-linkprop net1
LINK PROPERTY VALUE DEFAULT POSSIBLE
net1 speed 1000 -- --
net1 autopush -- -- --
net1 zone -- -- --
net1 duplex half half,full
net1 state unknown up up,down
net1 adv_autoneg_cap 1 1 1,0
net1 mtu 1500 1500 --
net1 flowctrl no bi no,tx,rx,bi
net1 adv_1000fdx_cap 1 1 1,0
net1 en_1000fdx_cap 1 1 1,0
net1 adv_1000hdx_cap 1 1 1,0
net1 en_1000hdx_cap 0 0 1,0
net1 adv_100fdx_cap 0 0 1,0
net1 en_100fdx_cap 0 0 1,0
net1 adv_10hdx_cap 0 0 1,0
net1 en_10hdx_cap 0 0 1,0
```
データリンクプロパティーの設定

リンクの速度機能や重複機能の設定は、en_*_capというラベルの付いた有効化速度プロパティーに手動で構成されます。たとえば、en_1000fdx_capはギガビット全重複機能のプロパティーであり、en_100hdx_capは100メガビット半重複機能のプロパティーです。これらの有効化速度プロパティーの設定は、ホストとそのリンクパートナーとの間で、対応する通知速度プロパティーによって通知されます。通知速度プロパティーには、adv_1000fdx_capやadv_100hdx_capのように、adv_*_capというラベルが付けられます。

通常、ある特定の有効化速度プロパティーと対応する通知プロパティーの設定は、同一になります。ただし、NICがPower Managementなどの一部の高度な機能をサポートしている場合は、それらの機能が、ホストとリンクパートナーとの間で実際に通知されるビット数の制限を設定する可能性があります。たとえば、Power Managementを使用する場合、adv_*_capプロパティーの設定がen_*_capプロパティーの設定のサブセットにすぎなくなる可能性があります。有効化速度プロパティーおよび通知速度プロパティーの詳細については、dladm(1M)のマニュアルページを参照してください。

▼ ダイレクトメモリーアクセスバイアードングを使用するように
e1000g ドライバを設定する方法

この手順と次の手順は、非公開プロパティーの構成方法を示します。どちらの手順も、e1000gドライバに固有のプロパティーに適用されます。しかし、ほかのNICドライバの非公開プロパティーを構成する際にも、この一般的な手順を使用できます。

ファイル転送などの一括トポフィックスでは通常、ネットワーク経由で大きなパケットの発送発信が必要となります。そのような場合に、e1000gドライバからより高いパフォーマンスを引き出すためには、DMAバイアードの自動的に使用するようにドライバを構成します。その場合、バイアードのフラグメントサイズのしきい値が定義されます。フラグメントサイズがこのしきい値を超えた場合は、伝送にDMAバイアードが使用されます。フラグメントサイズがこのしきい値より小さい場合は、bcopyモードが使用され、事前に割り当てられた伝送バッファにフラグメントデータがコピーされます。

このしきい値を設定するには、次の手順を実行します。

1 管理者になります。
   詳細は、「Oracle Solarisの管理: セキュリティーサービス」の「管理権限を取得する方法」を参照してください。

2 _tx_bcopy_threshold プロパティーに対して適切な設定を行います。
   # dladm set-linkprop -p _tx_bcopy_threshold=value e1000g-datalink
   このプロパティーでは、しきい値の有効な設定は、60から2048までの範囲です。
データリンクプロパティーの設定

注: 公開プロパティーの構成と同じく、非公開プロパティーの設定を変更する前に、インタフェースの unplumb も行う必要があります。

3 (省略可能) しきい値の新しい設定を確認します。

```
# dladm show-linkprop -p _tx_bcopy_threshold e1000g-datalink
```

▼ 割り込みレートを手動で設定する方法

e1000g ドライバによって割り込みが発行されるレートを制御するパラメータは、ネットワークやシステムのパフォーマンスにも影響を与えます。通常、パケットごとに割り込みを生成することによって、ネットワークパケットがスタックの上位層に配信されます。また、割り込みレートはデフォルトで、カーネル内の GLD 層によって自動的に調整されます。ただし、ネットワークトラフィックの状態によっては、このモードが望ましくない場合があります。この問題の詳細については、1996 年の USENIX 技術会議で発表されたこのドキュメント（http://www.stanford.edu/class/cs240/readings/mogul.pdf）を参照してください。したがって、特定の環境でより良いパフォーマンスを得るには、割り込みレートの手動設定が必要になります。

割り込みレートを定義するには、次のパラメータを設定します。

- `_intr_throttling_rate` は、ネットワークトラフィックの状態にかかわらず、割り込み表記の延長を決定します。
- `_intr_adaptive` は、割り込みスロットリングレートの自動チューニングを有効にするかどうかを決定します。デフォルトでは、このパラメータは有効になっていません。

1 管理者になります。
詳細は、『Oracle Solaris の管理・セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 必要に応じて、ドライバのプロパティーを変更するデバイスを特定します。

```
# dladm show-phys
```

3 割り込みスロットリングレートの自動チューニングを無効にします。

```
# dladm set-linkprop -p _intr_adaptive=0 e1000g-datalink
```

注: 割り込みスロットリングレートの自動チューニングが有効になっていると、パラメータ `_intr_throttling_rate` の既存の設定は一切無視されます。

4 データリンク上に構成されている IP インタフェースをすべて削除します。

第8章・データリンクの構成と管理
データリンクプロパティの設定

5 最小割り込み間レベルの設定を行います。
   
   # dladm set-linkprop -p _intr_throttling_rate=value e1000g-datalink

注 - intr_throttling_rate パラメータのデフォルト設定は、SPARC ベースシステムでは 550、x86 ベースシステムでは 260 です。最小割り込み間レベルを 0 に設定すると、割り込みスロットリングログリッが無効になります。

6 IP インタフェースを構成します。

7 (省略可能) しきい値の新しい設定を表示します。

例 8-9 DMA パイニングの構成と割り込みスロットリングレートの設定

この例では、e1000g NIC を備えた x86 ベースシステムを使用します。ドライバは、パケージの伝送に DMA パイニングを使用する方法と bcopy モードを使用する方法の切り替えを行うしきい値設定で構成されます。割り込みスロットリングレートの設定は変更されます。さらに、e1000g データリンクは、OS によって割り当てられたデフォルトのマシン名を使用しています。したがって、カスタマイズされた名前 net0 を参照することによってデータリンクの構成が実行されます。

# dladm show-phys
   LINK MEDIA STATE SPEED DUPLEX DEVICE
   net0 ether up 100Mb full e1000g0

# dladm show-linkprop -p _tx_bcopy_threshold net0
   LINK PROPERTY VALUE DEFAULT POSSIBLE
   net0 _tx_bcopy_threshold 512 512 --

# dladm show-linkprop -p _intr-throttling_rate
   LINK PROPERTY VALUE DEFAULT POSSIBLE
   net0 _intr-throttling_rate 260 260 --

# ipadm delete-ip net0
# dladm set-linkprop -p _tx_bcopy_threshold=1024 net0
# dladm set-linkprop -p _intr_adaptive=0 net0
# dladm set-linkprop -p _intr-throttling_rate=1024 net0

# ipadm create-ip net0
# ipadm create-addr -T static -a 10.10.1.2/24 net0/v4addr
# dladm show-linkprop -p _tx_bcopy_threshold=1024 net0
   LINK PROPERTY VALUE DEFAULT POSSIBLE
   net0 _tx_bcopy_threshold 1024 512 --

# dladm show-linkprop -p _intr-adaptive net0
   LINK PROPERTY VALUE DEFAULT POSSIBLE
   net0 _intr-adaptive 0 1 --

# dladm show-linkprop -p _intr-throttling_rate
   LINK PROPERTY VALUE DEFAULT POSSIBLE
   net0 _intr-throttling_rate 1024 260 --
データリンクに関するその他の構成タスク

この節は、動的再構成 (DR) の実行や STREAMS モジュールの操作など、dladm コマンドを使用することで単純化された、その他の一般的な構成手順について説明します。

▼ 動的再構成を使用してネットワークインタフェースカードを交換する方法

この手順は、動的再構成 (DR) をサポートするシステムにのみ適用されます。ここでは、ネットワークリンク構成とネットワークハードウェア構成を分離することで DR がいかに容易になったかを示します。DR の完了後にネットワークリンクを構成し直す必要はなくなりました。代わりに、取り外した NIC のリンク構成を転送し、それを交換した NIC に継承させるだけです。

始める前に  DR の実行手順はシステムのタイプによって異なります。まず、次を必ず完了してください。

- システムが DR をサポートしていることを確認します。
- アクティブなネットワーク構成プロファイルが DefaultFixed になっていることを確認します。システムのアクティブな NCP が DefaultFixed でない場合の DR の使用方法については、42 ページの「その他の Oracle Solaris ネットワーク技術を使用した NWAM の動作」の「動的再構成とネットワーク構成プロファイル」の節を参照してください。

注 - 次の手順は、データリンクの柔軟な名前の使用に特に関係する DR の側面にのみ言及しています。この手順には、DR を実行するための完全な手順は含まれていません。システムの適切な DR ドキュメントを参照する必要があります。

管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。
データリンクに関するその他の構成タスク

2 (省略可能) データリンクの物理属性や各データリンクのシステム上の場所に関する情報を表示します。

```bash
# dladm show-phys -L
```

`dladm show-phys -L` によって表示される情報のタイプの詳細については、`dladm(1M)` のマニュアルページを参照してください。

3 システムのドキュメントで詳しく説明されている DR 手順を実行し、NIC を取り外してから交換用の NIC を挿入します。
この手順を実行するには、システムの DR ドキュメントを参照してください。

交換用 NIC を取り付けたら、次の手順に進みます。

4 古い NIC 同じスロットに交換用 NIC を挿入した場合は、手順 6 に進みます。それ以外の場合は次の手順に進みます。

古い NIC が以前に占有していたと同じ場所を新しい NIC が使用しているため、新しい NIC が古い NIC のリンク名と構成を継承します。

5 該当する状況に応じて、次のいずれかの手順を実行します。

- 交換する古い NIC がシステムのスロット内に未使用 NIC として残っている場合は、次の手順を実行します。
  a. 交換する NIC に別の名前を割り当てます。

```bash
# dladm rename-link oldNIC new-name
```

`oldNIC` 交換されるがシステム内に残される NIC を表します。

`new-name` removedNIC に付ける新しい名前を表します。この名前は、システム内のほかのどのリンクとも共有してはいけません。

b. 交換用 NIC に古い NIC の名前を割り当てます。

```bash
# dladm rename-link replacementNIC oldNIC
```

`replacementNIC` 取り付けたばかりの新しい NIC を表します。この NIC は、システム内で占有しているスロットに応じたデフォルトリンク名を自動的に受け取ります。

`oldNIC` 交換されるがシステム内に残される NIC を表します。

- 古い NIC は取り外し、交換用 NIC を別のスロットに取り付けたが、その NIC に古い NIC の構成を継承させたい場合には、古い NIC の名前を新しい NIC に割り当てます。

```bash
# dladm rename-link replacementNIC oldNIC
```

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
データリンクに関するその他の構成タスク

6 新しいNICリソースがOracle Solarisによって使用可能になるようにして、DRプロセスを完了します。
たとえば、cfgadmコマンドを使用してNICを構成します。詳細は、cfgadm(1M)のマニュアルページを参照してください。

7 (省略可能)リンクの情報を表示します。
たとえば、dladm show-phys、dladm show-linkのいずれかを使用して、データリンクに関する情報を表示できます。

例 8-10 新しいネットワークカードを取り付けることによる動的再構成の実行
この例は、リンク名net0のbgeカードをe1000gカードに交換する方法を示します。
e1000gがシステムに接続されると、net0のリンク構成がbgeからe1000gに転送されます。

```
# dladm show-phys -L
LINK DEVICE LOCATION
net0 bge0 MB
net1 ibp0 MB/RISER0/PCIE0/PORT1
net2 ibp1 MB/RISER0/PCIE0/PORT2
net3 eob2 MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

cfgadmを使用してbgeを取り外してそこにe1000gを取り付けるなど、DR固有の手順を実行します。カードを取り付けると、e1000gのデータリンクは自動的に名前net0を引き継ぎ、そのリンク構成を継承します。

```
# dladm show-phys -L
LINK DEVICE LOCATION
net0 e1000g0 MB
net1 ibp0 MB/RISER0/PCIE0/PORT1
net2 ibp1 MB/RISER0/PCIE0/PORT2
net3 eob2 MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

# dladm show-link
```
LINK CLASS MTU STATE OVER
net0 phys 9600 up ---
net1 phys 1500 down ---
net2 phys 1500 down ---
net3 phys 1500 down ---
```

データリンクでのSTREAMSモジュールの構成
必要に応じて、データリンク上にブッシュされるSTREAMSモジュールを最大8個設定できます。これらのモジュールは通常、仮想プライベートネットワーク(VPN)やファイアウォールなど、他社製のネットワークソフトウェアによって使用されます。そのようなネットワークソフトウェアに関するドキュメントは、ソフトウェアベンダーから提供されています。
データリンクに関するその他の構成タスク

特定のデータリンク上にプッシュする STREAMS モジュールのリストは、autopush リンクプロパティによって制御されます。また、autopush リンクプロパティの値は、dladm set-linkprop サブコマンドを使用することによって設定されます。

また、独立した autopush コマンドを使用して STREAMS autopush モジュールをドライバ単位で設定することもできます。ただし、ドライバは常に NIC にバインドされます。データリンクのベースとなる NIC が取り外されると、そのリンクの autopush プロパティの情報も失われます。

データリンク上にプッシュされる STREAMS モジュールを構成する際は、autopush コマンドに優先して dladm set-linkprop コマンドを使用してください。特定のデータリンクでドライバ単位とリンク単位の両方のタイプの autopush 構成が存在している場合、dladm set-linkprop で設定されたリンク単位の情報が使用され、ドライバ単位の情報は無視されます。

▼ データリンクに STREAMS モジュールを設定する方法

次の手順は、dladm set-linkprop コマンドを使用して STREAMS モジュールを構成する方法について説明します。

1 管理者になります。
詳細は、「Oracle Solaris の管理: セキュリティーバイズ」の「管理権限を取得する方法」を参照してください。

2 リンクが開かれたときにモジュールをストリームにプッシュします。

```
# dladm set-linkprop -p autopush=modulelist link
```

`modulelist` 自動的にストリームにプッシュされるモジュールのリストを指定します。最大 8 個のモジュールをリンク上にプッシュできます。これらのモジュールは、`modulelist` に指定した順番でプッシュされます。ドットを区切り記号として使用して、リスト内のモジュールを区切ります。

`link` モジュールのプッシュ先となるリンクを指定します。

例 8–11 autopush リンクプロパティの設定

この例では、vpnmod および bufmod モジュールをリンク net0 の上にプッシュします。このリンクのベースとなるデバイスは、bge0 です。

```
# dladm set-linkprop -p autopush=vpnmod.bufmod net0
```

あとでこの bge カードを e1000g に交換する場合は、autopush の設定を構成し直さなくても新しいデータリンクに切り替えることができます。e1000g カードは、bge のリンク名と構成を自動的に継承します。
▼ autopush リンクプロバティーの設定を取得する方法

1 管理者になります。
詳細は、『Oracle Solaris の管理:セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 autopush リンクプロバティーの設定を表示します。
   # dladm show-linkprop -p autopush [link]
   link を指定しなかった場合、すべての構成済みリンクの情報が表示されます。

▼ autopush リンクプロバティーの設定を削除する方法

1 管理者になります。
詳細は、『Oracle Solaris の管理:セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 特定のデータリンクの autopush リンクプロバティーの設定を削除します。
   # dladm reset-linkprop [-t] -p autopush link
   プロバティーの設定を一時的に削除する場合は、-t オプションを使用します。シス
   テムのリブート時に設定が復元されます。
IP インタフェースの構成

この章では、データリンク上に IP インタフェースを構成するために使用される手順について説明します。

IP インタフェースの構成について

Oracle Solaris をインストールしたあと、次のタスクを実行する場合があります。

- 基本的なインタフェース構成のために、データリンク上に IP インタフェースを構成します。この章では、これらの手順について説明します。
- 無線インタフェースを構成します。これらの手順は、第 10 章「Oracle Solaris 上での無線インタフェース通信の構成」で説明されています。
- IPMP グループのメンバーとしての IP インタフェースを構成します。これらの手順は、第 15 章「IPMP の管理」で説明されています。

ipadm コマンド

Oracle Solaris における進捗は、ネットワーク構成のさまざまな側面を効率的に管理するための従来のツールの機能範囲を超えています。たとえば、ifconfig コマンドはこれまで、ネットワークインタフェースを構成するために慣例的に使用されるツールでした。ただし、このコマンドには、永続的な構成設定が実装されていません。ifconfig ではこれまで、ネットワーク管理における追加機能のための拡張機能が追加されてきました。ただし、その結果、このコマンドは複雑で、使いづらくなっています。

インタフェースの構成と管理に関する別の問題として、TCP/IP インターネットプロトコルのプロバディーまたはチューニング可能パラメータを管理するための簡単なツールが存在しない点があります。ndd コマンドは、この目的のための定番のカスタマイズツールでした。ただし、ifconfig コマンドと同様に、ndd には永続的な構成設
ipadmコマンド

特定が実装されていません。以前は、ブートスクリプトの編集によって、ネットワークのシナリオで永続的な設定をシミュレートすることが可能でした。Oracle SolarisのSMF機能の導入により、特に、Oracle Solarisインストールへのアップグレードを考慮した場合のSMF依存性の管理の複雑さのために、このような回避策の使用はリスクを伴うことがあります。

ipadmコマンドは、インタフェース構成のためのifconfigコマンドを最終的に置き換えるために導入されました。このコマンドはまた、プロトコルのプロパティを構成するためのnddコマンドも置き換えます。

インタフェースを構成するためのツールとして、ipadmコマンドには次の利点があります。

- インタフェース構成以外の目的にも使用されるifconfigコマンドとは異なり、IPインタフェースの管理に特化したツールであるため、IPインタフェースとIPアドレスをより効率的に管理します。
- インタフェースとアドレスの永続的な構築設定を実装するためのオプションを提供します。

ifconfigの各オプションとそれと同等のipadmサブコマンドの一覧については、204ページの「ifconfigコマンドのオプションとipadmコマンドのオプション」を参照してください。

プロトコルのプロパティを設定するためのツールとして、ipadmコマンドには次の利点があります。

- IP、アドレス解決プロトコル（ARP）、SCTP（Stream Control Transmission Protocol）、ICMP（Internet Control Messaging Protocol）のほか、TCPやユーザーデータグラムプロトコル（UDP）などの上位階層プロトコルの一時的または永続的なプロトコルのプロパティを設定できます。
- プロパティの現在およびデフォルトの設定や、指定可能な設定の範囲などの、各TCP/IPパラメータに関する情報を提供します。そのため、デバッグ情報の取得が容易になります。
- ipadmコマンドはまた、一貫性のあるコマンド構文に従っているため、より容易に使用できます。

nddの各オプションとそれと同等のipadmサブコマンドの一覧については、206ページの「nddコマンドのオプションとipadmコマンドのオプション」を参照してください。

ipadmコマンドの詳細は、ipadm(1M)のマニュアルページを参照してください。
IPインタフェースの構成(タスク)

この節では、IPインタフェースに関する基本的な構成手順について説明します。次の表は、各構成タスクについて説明し、これらのタスクを対応する手順にマップしています。

表9-1  IPインタフェースの構成(タスクマップ)

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>一意のMACアドレスをサポートするようにシステムを設定します。</td>
<td>インタフェースに対して一意のMACアドレスを許可するようにSPARCベースのシステムを構成します。</td>
<td>181ページの「SPARCインタフェースのMACアドレスが一意であることを確認する方法」</td>
</tr>
<tr>
<td>ipadmコマンドを使用して、基本的なIPインタフェース構成を実行します。</td>
<td>IPインタフェースを作成し、有効なIPアドレス(静的またはDHCP)を割り当てます。</td>
<td>183ページの「IPインタフェースを構成する方法」</td>
</tr>
<tr>
<td>ipadmコマンドを使用して、IPアドレスをカスタマイズします。</td>
<td>特定のIPアドレスのネットワークIDを設定します。</td>
<td>188ページの「IPアドレスのプロパティを設定する方法」</td>
</tr>
<tr>
<td>ipadmコマンドを使用して、インタフェース情報を取得します。</td>
<td>インタフェース、アドレス、およびプロトコルの各種のプロパティと、それらに対応する設定を一覧表示します。</td>
<td>198ページの「ネットワークインタフェースに関する情報を取得する方法」</td>
</tr>
</tbody>
</table>

▼SPARC: インタフェースのMACアドレスが一意であることを確認する方法

アプリケーションによっては、ホスト上のすべてのインタフェースでそれぞれ一意のMACアドレスが使用されている必要があります。ただし、すべてのSPARCベースのシステムは、システム共通MACアドレスを持っており、デフォルトではすべてのインタフェースがこのアドレスを使用します。次の2つの状況では、SPARCシステム上のインタフェースに出荷時に設定されたMACアドレスを構成する場合があります。

- リンク集約の場合、集約構成では出荷時に設定されたインタフェースのMACアドレスを使用してください。
- IPMPグループの場合、グループ内の各インタフェースで一意のMACアドレスを使用する必要があります。これらのインタフェースでは出荷時に設定されたMACアドレスを使用する必要があります。

EEPROMパラメータlocal-mac-address?によって、SPARCシステム上のすべてのインタフェースがシステム共通MACアドレスまたは一意のMACアドレスのどちらを
使用しているかを判別します。次の手順では、eepromコマンドを使用して、local-mac-address?の現在値をチェックし、必要に応じて変更する方法を示します。

1 管理者になります。
詳細は、『Oracle Solarisの管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 システム上のすべてのインタフェースがシステム共通MACアドレスを現在使用しているかどうかを判断します。
   # eeprom local-mac-address?
llocal-mac-address?=false
この例では、eepromコマンドの応答のlocal-mac-address?=falseによって、すべてのインタフェースがシステム共通MACアドレスを使用していることが示されています。local-mac-address?=falseの値は、インタフェースをIPMPグループのメンバーにする前に、local-mac-address?=trueに変更する必要があります。集約の場合にも、local-mac-address?=falseをlocal-mac-address?=trueに変更してください。

3 必要に応じて、local-mac-address?の値を次のように変更します。
   # eeprom local-mac-address?=true
システムをリブートすると、出荷時に設定されたMACアドレスを持ちインタフェースは、システム共通MACアドレスの代わりに、その出荷時の設定を使用します。出荷時に設定されたMACアドレスを持たないインタフェースは、システム共通MACアドレスを引き続き使用します。

4 システム上のすべてのインタフェースのMACアドレスをチェックします。
複数のインタフェースが同じMACアドレスを持つ場合がないかどうかを調べてください。この例では、すべてのインタフェースがシステム共通MACアドレス8:0:20:0:0:1を使用しています。
   # dladm show-linkprop -p mac-address

<table>
<thead>
<tr>
<th>LINK</th>
<th>PROPERTY</th>
<th>PERM</th>
<th>VALUE</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>net0</td>
<td>mac-address</td>
<td>rw</td>
<td>8:0:20:0:0:1</td>
<td>8:0:20:0:0:1</td>
<td>--</td>
</tr>
<tr>
<td>net1</td>
<td>mac-address</td>
<td>rw</td>
<td>0:14:4f:45:c:2d</td>
<td>0:14:4f:45:c:2d</td>
<td>--</td>
</tr>
</tbody>
</table>

注- 同じMACアドレスを持つ複数のネットワークインタフェースがまだ残っている場合だけ、次の手順を続けます。それ以外の場合は、最後の手順に進んでください。

5 すべてのインタフェースが一意のMACアドレスを持つように、必要に応じて、残りのインタフェースを手動で構成します。
   # dladm set-linkprop -p mac-address=mac-address interface
前の手順の例では、net0とnet1をローカルで管理されているMACアドレスで構成する必要がありました。たとえば、ローカルで管理されているMACアドレス06:05:04:03:02でnet0を再構成するには、次のコマンドを入力します。

```
# dladm set-linkprop -p mac-address=06:05:04:03:02 net0
```

このコマンドの詳細は、dladm(1M)のマニュアルページを参照してください。

6 システムをリブートします。

### IPインタフェースの構成

次の手順は、IP構成の各種のニーズのためにipadmコマンドを使用する方法を示しています。ifconfigコマンドも引き続きインターフェースの構成に使用できますが、ipadmコマンドの方が推奨されるツールです。ipadmコマンドとその利点の概要については、「IPインタフェースの構成と管理」を参照してください。

注 - 通常、IPインタフェースの構成とデータリンクの構成は同時に発生します。そのため、該当する場合、次の手順にはdladmコマンドを使用したデータリンクの構成手順が含まれます。dladmコマンドを使用してデータリンクを構成および管理する方法についての詳細は、「第8章「データリンクの構成と管理」を参照してください。

### ▼ IPインタフェースを構成する方法

次の手順は、IPインタフェースの基本的な構成を実行する例を示しています。

#### 始める前に

システム上のデータリンクの名前を変更するかどうかを決定します。通常は、データリンクにデフォルトで割り当てられている汎用名を使用します。リンク名を変更するには、「IPインタフェースの構成と管理」の「データリンクの名前を変更する方法」を参照してください。

1 管理者になります。

詳細は、「Oracle Solarisの管理:セキュリティサービス」の「管理権限を取得する方法」を参照してください。

2 (省略可能) 現在システム上に存在するデータリンクの物理的属性に関する情報を表示します。

```
# dladm show-phys
```

このコマンドは、システムにインストールされている物理ネットワークカードとそれらのプロパティのいくつかを示します。このコマンドについての詳細は、データリンクの物理属性に関する情報を表示する方法を参照してください。
IPインタフェースの構成

3 現在システム上に存在するデータリンクに関する情報を表示します。
   # dladm show-link
   このコマンドは、データリンクと、それに対して設定されている特定のプロパティー(リンクの作成に使用した物理カードを含む)を示します。

4 IPインタフェースを作成します。
   # ipadm create-interface-class interface
   interface-class ユーザーが作成できる次の3つのインタフェースクラスのうちの1つを示します。
   - IPインタフェース。このインタフェースクラスは、ネットワーク構成を実行するときに作成するもっとも一般的なクラスです。このインタフェースクラスを作成するには、create-ipサブコマンドを使用します。
   - STREAMS仮想ネットワークインタフェースドライバ(VNIインタフェース)。このインタフェースクラスを作成するには、create-vniサブコマンドを使用します。VNIデバイスまたはインタフェースの詳細は、vni(7d)のマニュアルページを参照してください。
   - IPMPインタフェース。このインタフェースは、IPMPグループを構成するときに使用されます。このインタフェースクラスを作成するには、create-ipmpサブコマンドを使用します。IPMPグループについての詳細は、第14章「IPMPの紹介」および第15章「IPMPの管理」を参照してください。

interface インタフェースの名前を示します。この名前は、インタフェースの作成に使用しているリンクの名前と同じです。

注 - IPインタフェースにIPアドレスを割り当てるには、そのIPインタフェースを事前に作成する必要があります。

5 IPインタフェースに有効なIPアドレスを構成します。
次の構文では、インタフェースに静的アドレスを割り当てます。IPアドレスを割り当てるためのその他のオプションについては、ipadm(1M)のマニュアルページを参照してください。
   # ipadm create-addr -T address-type -a address/prefixlen addrobj
   -T address-type インタフェースに割り当てられるIPアドレスのタイプを指定します。static、dhcp、addrconfのいずれかです。addrconfは、自動的に生成されたIPv6アドレスを示します。
   -a インタフェース上に構成するIPアドレスを指定します。ローカルアドレスのみか、またはトンネル構成の場合はローカルアドレス
とリモートアドレスの両方を指定できます。通常は、ローカルアドレスのみを割り当てます。この場合は、-aオプションでアドレスを直接指定します。たとえば、-a address とします。このアドレスは自動的にローカルアドレスとみなされます。

トンネルを構成している場合は、システムのローカルアドレスと宛先システムのリモートアドレスの両方の指定が必要になることがあります。この場合は、2つのアドレスを区別するためにlocalとremoteを次のように指定する必要があります：-a local=local-addr, remote=remote-addr。トンネルの構成についての詳細は、「Oracle Solarisの管理:IPサービス」の第6章「IPトンネルの構成」を参照してください。

数値IPアドレスを使用している場合は、CIDR表記のアドレスにaddress/prefixlenという形式を使用します。たとえば、1.2.3.4/24とします。prefixlenオプションの説明を参照してください。

必要に応じて、数値IPアドレスの代わりにaddressにホスト名を指定できます。ホスト名の使用が有効なのは、そのホスト名に対応する数値IPアドレスが/etc/hostsファイルで定義されている場合です。このファイルで数値IPアドレスが定義されていない場合は、name-service/switchサービスでhostに対して指定されているリソルバの順序を使用して、この数値が一意に取得されます。特定のホスト名に対して複数のエントリが存在する場合は、エラーが生成されます。

注-ブートプロセスにおいて、IPアドレスの作成は、ネームサービスがオンラインになる前に実行されます。そのため、ネットワーク構成で使用されているすべてのホスト名が/etc/hostsファイルで定義されているようにする必要があります。

(prefixlen) CIDR表記を使用する場合にIPv4アドレスの一部であるネットワークIDの長さを指定します。アドレス12.34.56.78/24では、24がprefixlenです。prefixlenを含まない場合は、name-service/switchサービスでnetmaskに対して示されているシーケンスに従って、またはクラスフルアドレスのセマンティクスを使用してネットマスクが計算されます。

(addrobject) システムで使用されている一意のIPアドレスまたはアドレスのセットの識別子を指定します。これらのアドレスは、IPv4またはIPv6タイプのいずれかです。この識別子は、interface/user_specified_stringという形式を使用します。
interface は、アドレスが割り当てられる IP インタフェースを示します。interface 変数には、IP インタフェースが構成されるデータリンクの名前が反映されている必要があります。

user-specified-string は、アルファベット文字で始まり、最大長が 32 文字の英数字の文字列を示します。その後、ipadm show-addr や ipadm delete-addr などの、システム内のアドレスを管理する任意の ipadm サブコマンドを使用するときに、数値 IP アドレスの代わりに addrobj を参照できます。

6 (省略可能) 新しく構成された IP インタフェースに関する情報を表示します。調べる対象の情報に応じて、次のコマンドを使用できます。

- インタフェースの一般的なステータスを表示します。
  
  ```
  # ipadm show-if [interface]
  ```

- インタフェースのアドレス情報の表示します。
  
  ```
  # ipadm show-addr [addrobj]
  ```

addrobj を指定しない場合は、システム内のすべてのアドレス情報が表示されます。

ipadm show-* サブコマンドの出力についての詳細は、198 ページの「IP インタフェースとアドレスの監視」を参照してください。

7 (省略可能) /etc/hosts ファイル内に IP アドレスのエントリを追加します。
このファイル内のエントリは、IP アドレスとそれに対応するホスト名で構成されています。

注 - この手順は、ホスト名を使用する静的 IP アドレスを構成している場合にのみ適用されます。DHCP アドレスを構成している場合は、/etc/hosts ファイルを更新する必要はありません。

例 9-1 静的アドレスによるネットワークインタフェースの構成

```bash
# dladm show-phys
LINK MEDIA STATE SPEED DUPLEX DEVICE
net3 Ethernet up 100Mb full bge3

# dladm show-link
LINK CLASS MTU STATE BRIDGE OVER
net3 phys 1500 up -- --
```
例9-2  
IP アドレスによるネットワークインタフェースの自動的な構成

この例では、前の例と同じネットワークデバイスを使用しますが、自身のアドレスを DHCP サーバーから受信するように IP インタフェースを構成します。

```
# ipadm create-ip net3
# ipadm create-addr -T static -a 192.168.84.24/24 net3/v4static

# ipadm show-if
+ IFNAME + CLASS + STATE + ACTIVE + OVER +
+ lo0 + loopback + ok + yes + -- +
+ net3 + ip + ok + yes + -- +

# ipadm show-addr
+ ADDROBJ + TYPE + STATE + ADDR +
+ lo0/7 + static + ok + 127.0.0.1/8 +
+ net3/v4 + static + ok + 192.168.84.3/24 +

# vi /etc/hosts
# Internet host table
# 127.0.0.1 localhost
10.0.0.14 myhost
192.168.84.3 campus01

/etc/hosts ファイルで campus01 がすでに定義されている場合は、次のアドレスを割り当てるときにそのホスト名を使用できます。

# ipadm create-addr -T static -a campus01 net3/v4static

# dladm show-phys
+ LINK + MEDIA + STATE + SPEED + DUPLEX + DEVICE +
+ net3 + Ethernet + up + 100Mb + full + bge3 +

# dladm show-link
+ LINK + CLASS + MTU + STATE + BRIDGE + OVER +
+ net3 + phys + 1500 + up + -- + -- +

# ipadm create-ip net3

# ipadm create-addr -T dhcp net3/dhcp

# ipadm show-if
+ IFNAME + CLASS + STATE + ACTIVE + OVER +
+ lo0 + loopback + ok + yes + -- +
+ net3 + ip + ok + yes + -- +

# ipadm show-addr net3/dhcp
+ ADDROBJ + TYPE + STATE + ADDR +
+ net3/dhcp + dhcp + ok + 10.8.48.242/24 +

# ipadm show-addr
+ ADDROBJ + TYPE + STATE + ADDR +
+ lo0/7 + static + ok + 127.0.0.1/8 +
+ net3/dhcp + dhcp + ok + 10.8.48.242/24 +
```
IPアドレスのプロパティーの設定

ipadmコマンドを使用すると、インタフェースにアドレスが割り当てられたあと、それらのアドレス固有のプロパティーを設定できます。これらのプロパティーを設定することによって、次のことを決定できます。

- アドレスのprefixlen。
- IPアドレスを送信パケットの発信元アドレスとして使用できるかどうか。
- アドレスが大域ゾーンまたは非大域ゾーンのどちらに属するか。
- アドレスがプライベートアドレスであるかどうか。

IPアドレスのプロパティーを一覧表示するには、次の構文を使用します。

```
# ipadm show-addrprop [-p property] [addrobj]
```

表示される情報は、使用するオプションによって異なります。

- プロパティーもアドレスオブジェクトも指定しない場合は、既存のすべてのアドレスのすべてのプロパティーが表示されます。
- プロパティーのみを指定した場合は、そのアドレスのそのプロパティーが表示されます。
- アドレスオブジェクトのみを指定した場合は、そのアドレスオブジェクトのすべてのプロパティーが表示されます。

注 - 一度に設定できるアドレスのプロパティーは1つだけです。

▼ IPアドレスのプロパティーを設定する方法

この手順は、IPアドレスのプロパティーを構成するための一般的な手順を示しています。

1 管理者になります。
詳細は、『Oracle Solarisの管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 システム上で現在使用されているIPアドレスを一覧表示します。
```
# ipadm show-addr
```

3 (省略可能)変更するIPアドレスの特定のプロパティーの現在の設定を判定します。
```
# ipadm show-addrprop -p property addrobj
```
プロパティーがわからない場合は、一般的なipadm show-addrpropコマンドを発行できます。このコマンドを使用してIPアドレスを表示すると、アドレスは、そのすべてのプロパティーの現在の設定とともに表示されます。
4 選択されたプロパティーを目的の値に設定します。
   # ipadm set-addrprop -p property=value addrobj

5 プロパティーの新しい設定を表示します。
   # ipadm show-addrprop -p property addrobj

例 9-3 アドレスの prefixlen プロパティーの設定

prefixlen プロパティーは、IP アドレスのネットマスクを示します。次の例では、net3 の IP アドレスの prefixlen プロパティーの長さを変更します。この例では、プロパティーの一時的な変更のみを作成するために -t オプションが使用されています。システムがリブートされると、このプロパティーの値はデフォルト設定に戻ります。

   # ipadm show-addr
   ADDR OBJ TYPE STATE ADDR
   lo0/- static ok 127.0.0.1/8
   net3/v4 static ok 192.168.84.3/24

   # ipadm show-addrprop -p prefixlen net3/v4
   ADDR OBJ PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
   net3/v4 prefixlen rw 24 24 24 1-30,32

   # ipadm set-addrprop -t -p prefixlen=8 net3/v4
   # ipadm show-addrprop -p prefixlen net3/v4
   ADDR OBJ PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
   net3/v4 prefixlen rw 8 24 24 1-30,32

IP インタフェースのプロパティーの設定

IP インタフェースには、データリンクと同様に、特定のネットワーク設定のためにユーザーがカスタマイズできるプロパティーがあります。インタフェースごとに、それぞれ IPv4 プロトコルと IPv6 プロトコルに適用される 2 つのプロパティーが設定されています。MTU などの一部のプロパティーは、データリンクと IP インタフェースの両方に共通です。そのため、データリンクに対してある MTU 設定を構築し、そのリンク上に構成されているインタフェースに対して別の MTU 設定を構築することができます。さらに、その IP インタフェースをたどり、それぞれ IPv4 パケットと IPv6 パケットに適用される異なる MTU 設定を構築できます。

IP 転送は、通常はネットワーキングのシナリオで構成される IP インタフェースのプロパティーです。次的手順は、この手順を示しています。

パケット転送の有効化

ネットワーク内で、あるホストは、別のホストシステムに宛てられたデータパケットを受信できます。受信側のローカルシステムでパケット転送を有効にするこ
IPインタフェースの構成

とによって、そのシステムは、データパケットを宛先ホストに転送できます。デフォルトでは、IP転送は無効になっています。次の2つの手順では、この機能を有効にする方法について説明します。以前のOracle Solarisリリースでは、パケット転送を有効にするためにrouteadmコマンドが使用されました。この手順でのipadm構文によってrouteadmコマンドが置き換えられました。

インターフェースベースの手順またはプロトコルベースの手順のどちらを使用するかを判断するには、次の点を考慮してください。

■ パケットの転送方法を選択できるようにする場合は、インタフェース上でパケット転送を有効にします。たとえば、複数のNICを備えたシステムがあるとします。一部のNICが外部ネットワークに接続されているのに対して、その他のNICはプライベートネットワークに接続されています。そのため、すべてのインタフェースではなく、一部のインタフェース上でのみパケット転送を有効にします。190ページの「インターフェースのプロパティを設定することによってIPパケット転送を有効にする方法」を参照してください。

■ システム内でグローバルにパケット転送を実装する場合は、プロトコルのforwardingプロパティを有効にします。この2番目の方法については、192ページの「プロトコルのプロパティを設定することによってパケット転送を有効にする方法」を参照してください。

注 - このパケット転送の2つの方法は、相互に排他的ではありません。たとえば、グローバルにパケット転送を有効にしたあと、インタフェースごとにforwardingプロパティをカスタマイズできます。そのため、その特定のシステムでは、パケット転送は引き続き選択可能です。

▼ インターフェースのプロパティを設定することによってIPパケット転送を有効にする方法

この手順は、特定のインタフェース上でIP転送のプロパティを構成することによって選択的にパケット転送を有効にする方法を示しています。

注 - パケット転送にはIPプロトコルが必要です。また、そのため、この手順にはIPのプロトコルバージョンの区別も含まれます。

1 管理者になります。
詳細は、「Oracle Solarisの管理:セキュリティサービス」の「管理権限を取得する方法」を参照してください。

2 インターフェースのIP転送のプロパティの現在の設定を表示します。

```
# ipadm show-ifprop -p forwarding [-m protocol-version] interface
```
ここで、protocol-version は ipv4 または ipv6 のどちらかです。バージョンを指定しない場合は、IPV4 プロトコルとIPV6 プロトコルの両方の設定が表示されます。

注-特定のインタフェースの有効なプロトコルのプロバティーをすべて表するには、次に示すように、プロバティーを指定しないでください。

# ipadm show-ifprop interface
この構文は、例9-4にも示されています。

3 パケット転送を有効にするインタフェースごとに、次のコマンドを入力します。

# ipadm set-ifprop forwarding=on -m protocol-version interface

4 (省略可能)インタフェースの forwarding プロバティーの設定を表示します。

# ipadm show-ifprop -p forwarding interface

5 インタフェースの forwarding プロバティーをデフォルト設定に戻すには、次のコマンドを入力します。

# ipadm reset-ifprop -p forwarding -m protocol-version interface

例9-4 インタフェースでの IPv4 パケットのみの転送を有効にする

次の例は、選択的なパケット転送を実装する方法を示しています。ここでは、net0 インタフェースでのみ IPv4 パケットの転送が有効になります。システムのほかの残りのインタフェースでは、パケット転送は無効(デフォルト設定)になっています。

# ipadm show-ifprop -p forwarding net0

- p property オプションを使用する ipadm show-ifprop コマンド構文では、特定のプロバティーに関する情報のみが表示されます。

# ipadm set-ifprop -p forwarding=on -m ipv4 net0
# ipadm show-ifprop net0

- p property オプションを使用しない ipadm show-ifprop コマンド構文では、インタフェースのすべてのプロバティーとそれらに対応する設定が表示されます。

# ipadm reset-ifprop -p forwarding -m ipv4 net0
# ipadm show-ifprop -p forwarding -m ipv4 net0
### IPインタフェースの構成

<table>
<thead>
<tr>
<th>net0</th>
<th>forwarding</th>
<th>ipv4</th>
<th>rw</th>
<th>off</th>
<th>off</th>
<th>off</th>
<th>on</th>
<th>off</th>
</tr>
</thead>
</table>

ipadm reset-ifprop コマンドは、指定されたプロパティーをデフォルト設定にリセットします。

### プロトコルのプロパティーを設定することによってパケット転送を有効にする方法

この手順は、システム内でグローバルにパケット転送を有効にする方法を示しています。

1. **管理者になります。**
   
   詳細は、「Oracle Solaris の管理: セキュリティーサービス」の「管理権限を取得する方法」を参照してください。

2. **IP転送のプロパティーの現在の設定を表示します。**
   
   ```bash
   # ipadm show-prop -p forwarding protocol-version
   ```
   
   ここで、protocol-version は ipv4 または ipv6 のどちらかです。

注: 特定のプロトコルの有効なすべてのチューニング可能パラメータのプロパティーと、それらの現在の設定を表示するには、次のコマンドを入力します。

```bash
# ipadm show-prop protocol
```

ここで、protocol は、ip、ipv4、ipv6、udp、tcp、icmp、および sctp のいずれかです。

この構文は、例 9-5 に示されています。

3. **転送を有効にするプロトコルバージョンごとに、次のコマンドを入力します。**
   
   ```bash
   # ipadm set-prop forwarding=on protocol-version
   ```

4. **(省略可能) 次のいずれかを実行することによって、IP転送のプロパティーの設定を表示します。**
   
   - プロトコルのすべてのプロパティーとプロトコルの現在の設定を表示するには、次のように入力します。
     ```bash
     # ipadm show-prop protocol
     ```
   - 特定のプロトコルの特定のプロパティーを表示するには、次のように入力します。
     ```bash
     # ipadm show-prop -p property protocol
     ```
   - 特定のプロトコルバージョンの特定のプロパティーを表示するには、次のように入力します。
     ```bash
     # ipadm show-prop -p property protocol
     ```
# ipadm show-prop -p property protocol-version

プロトコルバージョンの特定のプロパティをデフォルト設定にリセットするには、次のように入力します。

# ipadm reset-prop -p property protocol-version

例9-5 IPv4パケットとIPv6パケットの転送を有効にする

次の例は、インタフェース上でのパケットの転送に関する前の例に相当します。`ipadm show-prop`の2つの使用によって、指定されたプロパティの設定、またはプロトコルのすべてのプロパティとそれらに対応する設定が表示されます。

```bash
# ipadm show-prop -p forwarding ip
PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
ipv4 forwarding rw off -- off on,off
ipv6 forwarding rw off -- off on,off
#
# ipadm set-prop -p forwarding=on ipv4
# ipadm set-prop -p forwarding=on ipv6
#
# ipadm show-prop ip
PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
ipv4 forwarding rw on on off on,off
ipv4 ttl rw 255 -- 255 1-255
ipv6 forwarding rw on on off on,off
ipv6 hoplimit rw 255 -- 255 1-255#
```

プロトコルのプロパティの管理

インタフェースは別にして、ipadmコマンドを使用すると、プロトコルのプロパティ（チューニング可能パラメータとも呼ばれる）を構成できます。ipadmは、チューニング可能パラメータを設定するために以前のリリースで一般的に使用されていたnddコマンドを置き換えます。この節では、選択されたTCP/IPプロトコルのプロパティをカスタマイズするための手順と例について説明します。

TCP/IPプロパティの設定

TCP/IPプロパティは、インタフェーススペースまたはグローバルのどちらにもできます。各プロパティを特定のインタフェースに適用することも、ゾーン内のすべてのインタフェースにグローバルに適用することもできます。グローバルなプロパティは、非大域ゾーンごとに異なる設定にすることができます。サポートされるプロトコルのプロパティの一覧については、`ipadm(1M)`のマニュアルページを参照してください。
ネットワークが機能するには、通常、TCP/IP インターネットプロトコルのデフォルト設定で十分です。ただし、デフォルト設定が特定のネットワークトポロジーに対して十分でない場合は、次の表の順に、これらの TCP/IP プロパティをカスタマイズする方法を示しています。

この表は、プロトコルの特定のプロパティを構成するためのタスクと、それに対応する手順へのリンクを示しています。

表9-2 選択された TCP/IP プロパティの設定

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>ポートを特権ポートとしてマークします。</td>
<td>インタフェースのポートを、root ユーザー以外からのアクセスを制限するように予約します。</td>
<td>194 ページの「ポートのアクセスを root ユーザーのみに制限する方法」</td>
</tr>
<tr>
<td>マルチホームホスト上で受けまたは送信されている IP パケットの動作をカスタマイズします。</td>
<td>マルチホームホストで対称ルーティングをカスタマイズします。</td>
<td>196 ページの「マルチホームホスト上で対称ルーティングを実装する方法」</td>
</tr>
<tr>
<td>プロトコルのプロパティに関する情報を表示します。</td>
<td>プロトコルのプロパティとその現在の設定を表示します。</td>
<td>198 ページの「IP インタフェースとアドレスの監視」</td>
</tr>
</tbody>
</table>

注 - ipadmツールを使用してネットワークインタフェースとIPアドレスを構成する手順については、183 ページの「IP インタフェースの構成」を参照してください。

▼ ポートのアクセスを root ユーザーのみに制限する方法

TCP、UDP、SCTP などのトランスポートプロトコルでは、ポート1–1023はデフォルトの特権ポートです。つまり、root ユーザーのアクセス権で実行されるプロセスのみがこれらのポートにバインドできます。ipadm コマンドを使用することにより、この指定されたデフォルト範囲を超えて、あるポートを特権ポートにするように予約することができます。それによって、root プロセスのみがそのポートにバインドできます。この手順では、トランスポートプロトコルの次のプロパティを使用します。

- smallest_nonpriv_port
- extra_priv_ports

1 指定されたポートが通常のポートの範囲内にあって使用可能かどうかを判定します。

```
# ipadm show-prop -p smallest_nonpriv_port protocol
```
ここで、protocolは、特権ポートを構成する対象のプロトコルタイプです（IP、UDP、ICMPなど）。

このコマンド出力で、POSSIBLEフィールドは、通常のユーザーがバインドできるポート番号の範囲を示します。指定されたポートがこの範囲内にある場合は、そのポートを特権ポートとして設定できます。

2 予約しようとしているポートが使用可能であり、まだ特権ポートとしてマークされていないことを確認します。

```bash
# ipadm show-prop -p extra_priv_ports protocol
```

このコマンド出力で、CURRENTフィールドは、どのポートが現在特権ポートとしてマークされているかを示します。指定されたポートがこのフィールドに含まれていない場合は、そのポートを特権ポートとして設定できます。

3 指定されたポートを特権ポートとして追加します。

```bash
# ipadm set-prop -p extra_priv_ports=port-number protocol
```

4 特権ポートとして追加または削除するその他のポートごとに、次のいずれかを繰り返します。

- ポートを特権ポートとして追加するには、次の構文を入力します。

```bash
# ipadm set-prop -p extra_priv_ports+=portnumber protocol
```

注 - プラス記号 (+) の修飾子により、複数のポートを特権ポートにするように割り当てることができます。プラス記号の修飾子を使用すると、これらのポートの一覧を作成できます。この一覧にポートを個別に追加するには、この構文をこの修飾子とともに使用します。この修飾子を使用しない場合は、割り当てるポートにより、以前に特権ポートとして一覧表示されていたほかのすべてのポートが置き換えられます。

- ポートを特権ポートとして削除するには、次の構文を入力します。

```bash
# ipadm set-prop -p extra_priv_ports-=portnumber protocol
```

注 - マイナス記号 (-) の修飾子を使用すると、現在特権ポートとして一覧表示されている既存のポートからポートを削除できます。この同じ構文を使用して、ほかのすべての特権ポート（デフォルトポートを含む）を削除します。

5 指定されたポートの新しいステータスを確認します。

```bash
# ipadm show-prop -p extra_priv_ports protocol
```

このコマンド出力で、指定されたポートが現在CURRENTフィールドに含まれていることを確認します。
例9-6 特権ポートの設定

この例では、ポート 3001 と 3050 を特権ポートとして設定します。また、現在特権ポートとして一律表示されているポート 4045 を削除します。

smallest_nonpriv_port プロパティの出力で、POSSIBLE フィールドはポート 1024 が最小の非特権ポートであり、指定されたポート 3001 と 3050 が使用できる非特権ポートの範囲内にあることを示します。extra_priv_ports プロパティの出力で、ポート 2049 と 4045 は CURRENT フィールドで特権ポートとしてマークされています。そのため、ポート 3001 を特権ポートとして設定する処理に進むことができます。

```
# ipadm show-prop -p smallest_nonpriv_port tcp
PROTO PROPERTY   PERM CURRENT PERSISTENT DEFAULT POSSIBLE
tcp smallest_nonpriv_port rw 1024 -- 1024 1024-32768

# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY   PERM CURRENT PERSISTENT DEFAULT POSSIBLE
tcp extra_priv_ports rw 2049,4045 -- 2049,4045 1-65535

# ipadm set-prop -p extra_priv_ports+=3001 tcp
# ipadm set-prop -p extra_priv_ports+=3050 tcp

# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY   PERM CURRENT PERSISTENT DEFAULT POSSIBLE
tcp extra_priv_ports rw 2049,3001 3001,3050 2049,4045 1-65535
  3050

# ipadm set-prop -p extra_priv_ports-=4045 tcp
# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY   PERM CURRENT PERSISTENT DEFAULT POSSIBLE
tcp extra_priv_ports rw 2049,3001 3001,3050 2049,4045 1-65535
  3050
```

▼ マルチホームホスト上で対称ルーティングを実装する方法

デフォルトでは、複数のインタフェースを備えたシステム（マルチホームホストとも呼ばれる）は、経路指定テーブル内のトラフィックの宛先への一致する最長のルートに基づいてネットワークトラフィックを経路指定します。宛先への等しい長さの複数のルートが存在する場合、Oracle Solaris では、これらのルートにわたってトラフィックを分散させるために等コストマルチパス (ECMP) アルゴリズムを適用します。

この方法でのトラフィックの分散が、場合によっては最適ではないことがあります。ある IP パケットが、そのパケット内の IP 発信元アドレスと同じサブネット上に存在しないマルチホームホスト上のインタフェース経由で送信される可能性があります。さらに、送信パケットが特定の受信要求への応答（ICMP エコー要求など）である場合は、要求と応答が同じインタフェースをたどらない可能性があります。トラフィックのこのようなルーティング構成を非対称ルーティングと呼びます。インターネットサービスプロバイダーが RFC3704 ([http://rfc-editor.org/rfc/bcp/](http://rfc-editor.org/rfc/bcp/))
プロトコルのプロパティの管理

bcp84.txt)に記載されているインタフェースフィルタリングを実装している場合は、非対称ルーティング構成のために、送信パケットがプロバイダによって破棄されることがあります。

RFC 3704は、インターネット全体にわたるサービス拒否攻撃の制限を目的にしています。この目的に従うには、ネットワークを対称ルーティングに構成する必要があります。Oracle Solarisでは、IPのhostmodelプロパティを使用して、この要件を満たすことができます。このプロパティは、マルチホームホスト経由で受信また送信されているIPパケットの動作を制御します。

次の手順は、ipadmコマンドを使用して、特定のルーティング構成のためのhostmodelプロパティを設定する方法を示しています。

1 マルチホームホスト上で、管理者になります。

2 システム内のネットワークパケットのルーティングを構成します。

   `# ipadm set-prop -p hostmodel=value protocol`
   このプロパティは、次の3つの設定のいずれかに構成できます。

   - strong: RFC 1122で定義されている強い終端システム(ES)モデルに対応します。この設定によって、対称ルーティングが実装されます。
   - weak: RFC 1122で定義されている弱いESモデルに対応します。この設定では、マルチホームホストは非対称ルーティングを使用します。
   - src-priority: 優先ルートを使用してパケットルーティングを構成します。経路指定テーブル内に複数の宛先ルートが存在する場合、優先ルートは、送信パケットのIP発信元アドレスが構成されているインタフェースを使用するルートです。このようなルートが存在しない場合、送信パケットは、そのパケットのIP宛先への一致する最長のルートを使用します。

3 (省略可能)hostmodelプロパティの設定を確認します。

   `# ipadm show-prop protocol`

例9-7 マルチホームホスト上での対称ルーティングの設定

この例では、マルチホームホスト内のすべてのIPトラフィックの対称ルーティングを使用します。

   `# ipadm set-prop -p hostmodel=strong ip`
   `# ipadm show-prop -p hostmodel ip`

<table>
<thead>
<tr>
<th>PROTO</th>
<th>PROPERTY</th>
<th>PERM</th>
<th>CURRENT</th>
<th>PERSISTENT</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv6</td>
<td>hostmodel</td>
<td>rw</td>
<td>strong</td>
<td>--</td>
<td>weak</td>
<td>strong,</td>
</tr>
<tr>
<td>ipv6</td>
<td>src-priority</td>
<td>rw</td>
<td>strong</td>
<td>--</td>
<td>weak</td>
<td>src-priority, weak</td>
</tr>
<tr>
<td>ipv6</td>
<td>src-priority</td>
<td>rw</td>
<td>strong</td>
<td>--</td>
<td>weak</td>
<td>src-priority, weak</td>
</tr>
</tbody>
</table>

第9章・IPインタフェースの構成  197
**IPインタフェースとアドレスの監視**

*ipadm*コマンドはまた、IPインタフェースとそのプロバティーまたはパラメータを監視し、それらに関する情報を取得するための推奨されるツールでもあります。インタフェース情報を取得するための*ipadm*サブコマンドでは、次の基本的な構文が使用されます。

```
ipadm show-* [other-arguments] [interface]
```

- インタフェース情報を取得するには、*ipadm show-if* を使用します。
- アドレス情報を取得するには、*ipadm show-addr* を使用します。
- 特定のインタフェースのプロバティーに関する情報を取得するには、*ipadm show-ifprop* を使用します。
- 特定のアドレスのプロバティーに関する情報を取得するには、*ipadm show-addrprop* を使用します。

この節では、*ipadm* コマンドを使用してネットワークインタフェースに関する情報を取得するいくつかの例について説明します。ネットワーク上で実行するその他のタイプの監視タスクについては、『Oracle Solaris の管理: IP サービス』の第5章「TCP/IP ネットワークの管理」を参照してください。

注 - *ipadm show-* コマンド内のすべてのフィールドの説明については、*ipadm(1M)* のマニュアルページを参照してください。

▼ ネットワークインタフェースに関する情報を取得する方法

この手順では、インタフェースの一般的なステータス、アドレス情報、およびIPプロバティーに関する情報を表示する方法について説明します。

1. 管理者になります。
   詳細は、『Oracle Solaris の管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2. インタフェースに関するステータス情報を取得するには、次のコマンドを入力します。
   ```
   # ipadm show-if [interface]
   ```
インタフェースを指定しない場合は、この情報にシステム上のすべてのインターフェースが含まれます。

コマンド出力の各フィールドは、次の内容を示します。

**IFNAME**  情報が表示されているインタフェースを示します。

**CLASS**  インタフェースのクラスを示します。次の4つのいずれかです。
  - ip は IP インタフェースを示します。
  - ipmp は IPMP インタフェースを示します。
  - vni は仮想インタフェースを示します。
  - loopback は、自動的に作成されるループバックインタフェースを示します。ループバックインタフェースを除き、残りの3つのインタフェースクラスは手動で作成できます。

**STATE**  インタフェースのステータスを示します。ok、offline、failed、down、disabled のいずれかです。

failed のステータスは IPMP グループに適用され、ダウンしているためにトラフィックをホストできないデータリンクまたは IP インタフェースを示す場合があります。IP インタフェースが IPMP グループに属している場合、IPMP インタフェースは、グループ内のほかのアクティブな IP インタフェースを使用してトラフィックを引き続き送受信できます。

down のステータスは、管理者によってオフラインに切り替えられた IP インタフェースを示します。

disable のステータスは、ipadm disable-if コマンドを使用して unplumb されている IP インタフェースを示します。

**ACTIVE**  インタフェースがトラフィックをホストするために使用されているかどうかを示し、yes または no のどちらかに設定されます。

**OVER**  インタフェースの IPMP クラスにのみ適用され、IPMP インタフェースまたはグループを構成するベースとなるインタフェースを示します。

3 インタフェースのアドレス情報を取得するには、次のコマンドを入力します。

```
# ipadm show-addr [addrobj]
```

アドレス識別子を指定しない場合は、システム上のすべてのアドレス識別子に対するアドレス情報が表示されます。

コマンド出力の各フィールドは、次の内容を示します。

**ADDROBJ**  アドレスが表示されているアドレスオブジェクトを指定します。
4 インタフェースのプロパティーに関する情報を取得するには、次のコマンドを入力します。

```
# ipadm show-ifprop [-p property] interface
```

プロパティーを指定しない場合は、すべてのプロパティーとその設定値が表示されます。

コマンド出力の各フィールドは、次の内容を示します。

<table>
<thead>
<tr>
<th>TYPE</th>
<th>IP アドレスが static、dhcp、addrconf のいずれであるかを示します。addrconf の設定は、アドレスがステートレスまたはステートフルアドレス構成を使用して取得されたことを示します。</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATE</td>
<td>実際のアクティブな構成内のアドレスオブジェクトを示します。これらの値の完全な一覧については、ipadm(1M) のマニュアルページを参照してください。</td>
</tr>
<tr>
<td>ADDR</td>
<td>インタフェース上に構成されている IP アドレスを指定します。このアドレスは IPv4 または IPv6 のどちらかです。トンネルインタフェースには、ローカルアドレスとリモートアドレスの両方が表示されます。トンネルについての詳細は、『Oracle Solaris の管理: IP サービス』の第 6 章「IP トンネルの構成」を参照してください。</td>
</tr>
</tbody>
</table>

注 - いずれかのフィールド値が不明の場合（たとえば、情報を要求されているプロパティーがインタフェースでサポートされていない場合）、その設定は疑問符 (?) として表示されます。
アドレスのプロバティーに関する情報を取り出すには、次のコマンドを入力します。

```
# ipadm show-addrprop [-p property,...] [addrobj]
```

表示される情報は、使用するオプションによって異なります。

- プロバティーを指定しない場合は、すべてのプロバティーが表示されます。
- プロバティーのみを指定した場合は、すべてのアドレスのそのプロバティーが表示されます。
- アドレスオブジェクトのみを指定した場合は、システム上の既存のすべてのアドレスのプロバティーが表示されます。

コマンド出力の各フィールドは、次の内容を示します。

| ADDROBJ | プロバティーが表示されているアドレスオブジェクトを示します。 |
| PROPERTY | アドレスオブジェクトのプロバティーを示します。アドレスオブジェクトには、複数のプロバティーが含まれる場合があります。 |
| PERM | 指定されたプロバティーの許可されるアクセス権を示します。読み取り専用、書き込みのみ、またはその両方のいずれかです。 |
| CURRENT | 現在の構成内のプロバティーの実際の設定を示します。 |
| PERSISTENT | システムがリブートされたときに再適用されるプロバティーの設定を示します。 |
| DEFAULT | 指定されたプロバティーのデフォルト設定を示します。 |
| POSSIBLE | 指定されたプロバティーに割り当けることのできる設定の一覧を示します。数値の設定の場合は、受け入れ可能な範囲が表示されます。 |

例 9-8 ipadm コマンドを使用したインタフェースの監視

この一連の例は、ipadm show-* サブコマンドを使用して取得できる情報の種類を示しています。最初に、一般的なインタフェース情報が表示されます。次に、アドレス情報が表示されます。最後に、インタフェース net1 の MTU という特定のプロバティーに関する情報が表示されます。これらの例には、トンネルインタフェースや、カスタマイズされた名前を使用するインタフェースが含まれています。

```
# ipadm show-if
IFNAME  CLASS  STATE  ACTIVE  OVER
lo0     loopback  ok  yes  --
net0    ip  ok  yes  --
net1    ip  ok  yes  --
tun0    ip  ok  yes  --
```

```
# ipadm show-addr
ADDROBJ  TYPE  STATE  ADDR
```

第9章・IPインタフェースの構成 201
インターフェース構成のトラブルシューティング

この節では、ipadmコマンドを使用してIP インタフェースを構成している間に発生する可能性のある一般的な問題について説明します。

ipadm コマンドが動作しない。

dladmおよびipadmコマンドによる手動のIP インタフェース構成は、DefaultFixedなどの、固定タイプのネットワーク構成プロファイル(NCP)でのみ機能します。システム内のアクティブなNCPが自動タイプのプロファイルである場合は、dladmおよびipadmコマンドを使用する前に、固定タイプのプロファイルに切り替えてください。

# netadm list
TYPE PROFILE STATE
ncp DefaultFixed disabled
ncp Automatic online
loc Automatic offline
loc NoNet offline
...

# netadm enable -p ncp defaultfixed
ipadm create-addr コマンドでIPアドレスを割り当てることができない。

従来のifconfigコマンドでは、1つのコマンド構文でIPアドレスをplumbして割り当てることができます。ipadm.create-addrコマンドを使用してIPアドレスを構成する場合は、最初に別のコマンドでIPインタフェースを作成する必要があります。

```
# ipadm create-ip interface
# ipadm create-addr -T addr-type -a address addrobj
```

IPアドレスの構成中にcannot create address object: Invalid argument providedというメッセージが表示される。

アドレスオブジェクトは、IPインタフェースにバインドされた特定のIPアドレスを識別します。アドレスオブジェクトは、IPインタフェース上のIPアドレスごとの一意の識別子です。同じIPインタフェースに割り当てる2番目のIPアドレスを識別するために、別のアドレスオブジェクトを指定する必要があります。同じアドレスオブジェクト名を使用する場合は、別のIPアドレスを識別するために、アドレスオブジェクトの最初のインスタンスを削除してから割り当てる必要があります。

```
# ipadm show-addr
ADDROBJ TYPE STATE ADR
lo0 static ok 127.0.0.1/10
net0/v4 static ok 192.168.10.1

# ipadm create-addr -T static -a 192.168.10.5 net0/v4b
```

または

```
# ipadm show-addr
ADDROBJ TYPE STATE ADR
lo0 static ok 127.0.0.1/10
net0/v4 static ok 192.168.10.1

# ipadm delete-addr net0/v4
# ipadm create-addr -T static -a 192.168.10.5 net0/v4
```
IP インタフェースの構成中に cannot create address: Persistent operation on temporary object というメッセージが表示される。

ipadm コマンドは永続的な構成を作成します。構成している IP インタフェースが一時的なインタフェースとして作成された場合は、ipadm コマンドを使用して、そのインタフェース上に永続的な設定を構成することはできません。構成しているインタフェースが一時的であることを確認したあと、そのインタフェースを削除し、永続オプジェクトとして再作成してから、構成を再開してください。

```
# ipadm show-if -o all
IFNAME CLASS STATE ACTIVE CURRENT PERSISTENT OVER
lo0 loopback ok yes -m46-v------ 46-- --
net0 ip ok yes bm4-------- ---- --
```

PERSISTENT フィールドに 4 フラグ (IPv4 構成の場合) または 6 フラグ (IPv6 構成の場合) が存在しない場合は、net0 が一時的なインタフェースとして作成されたことを示します。

```
# ipadm delete-ip net0
# ipadm create-ip net0
## ipadm create-addr -T static -a 192.168.1.10 net0/v4
```

比較表：ipadm コマンドとその他のネットワークコマンド

ipadm コマンドは、IP インタフェース上のすべての構成タスクに使用するための推奨されるツールです。このコマンドは、ifconfig コマンドや ndd コマンドなどのネットワーク構成に使用されていた以前のリリースのコマンドを置き換えます。次表は、これらの以前のツールの選択されたコマンドオプションと、ipadm コマンドでのそれと同等のオプションを示しています。

注：これらの表には、ipadm のオプションの完全な一覧は示されていません。完全な一覧については、ipadm(IM) のマニュアルページを参照してください。

ifconfig コマンドのオプションと ipadm コマンドのオプション

次の表は、ifconfig コマンドのオプションと、ほぼ対応する ipadm サブコマンドを示しています。
<table>
<thead>
<tr>
<th><code>ifconfig</code>コマンド</th>
<th><code>ipadm</code>コマンド</th>
</tr>
</thead>
<tbody>
<tr>
<td>plumb/unplumb</td>
<td>ipadm create-ip</td>
</tr>
<tr>
<td></td>
<td>ipadm create-vni</td>
</tr>
<tr>
<td></td>
<td>ipadm create-ipmp</td>
</tr>
<tr>
<td></td>
<td>ipadm enable-addr</td>
</tr>
<tr>
<td></td>
<td>ipadm delete-ip</td>
</tr>
<tr>
<td></td>
<td>ipadm delete-vni</td>
</tr>
<tr>
<td></td>
<td>ipadm delete-ipmp</td>
</tr>
<tr>
<td></td>
<td>ipadm disable-addr</td>
</tr>
<tr>
<td>[address[/prefix-length] [dest-address]]</td>
<td>ipadm create-addr -T static</td>
</tr>
<tr>
<td>[addif address[ prefix-length]]</td>
<td>ipadm create-addr -T dhcp</td>
</tr>
<tr>
<td>[removeif address[ prefix-length]]</td>
<td>ipadm create-addr -T addrconf</td>
</tr>
<tr>
<td>[netmask mask]</td>
<td>ipadm show-addr</td>
</tr>
<tr>
<td>[destination dest-address]</td>
<td>ipadm delete-addr</td>
</tr>
<tr>
<td>[auto-dhcp</td>
<td>dhcp][primary][wait seconds]extend</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>[deprecated</td>
<td>-deprecated]</td>
</tr>
<tr>
<td>[preferred</td>
<td>-preferred]</td>
</tr>
<tr>
<td>[private</td>
<td>-private]</td>
</tr>
<tr>
<td>[zone zonename</td>
<td>-zones</td>
</tr>
<tr>
<td>up</td>
<td>ipadm up-addr</td>
</tr>
<tr>
<td>down</td>
<td>ipadm down-addr</td>
</tr>
<tr>
<td>[metric n] [mtu n] [nud</td>
<td>-nud] [arp</td>
</tr>
<tr>
<td>[usesrc [name</td>
<td>none] [router</td>
</tr>
<tr>
<td></td>
<td>ipadm reset-ifprop</td>
</tr>
<tr>
<td>[ipmp]</td>
<td>ipadm create-mpmp</td>
</tr>
<tr>
<td>[group [name</td>
<td>&quot;&quot;] standby</td>
</tr>
<tr>
<td>[failover</td>
<td>-failover]</td>
</tr>
<tr>
<td></td>
<td>ipadm remove-ipmp</td>
</tr>
<tr>
<td></td>
<td>ipadm set-ifprop -p [standby] [group]</td>
</tr>
</tbody>
</table>
### 比較表: ifconfig コマンドとipadm コマンドの構文の対応 (続き)

<table>
<thead>
<tr>
<th>ifconfig コマンド</th>
<th>ipadm コマンド</th>
</tr>
</thead>
<tbody>
<tr>
<td>[tdst tunnel-dest-addr] [tsrc tunnel-srsc-addr]</td>
<td>dladm *-iptun コマンドセット。詳細は、dladm(1M) のマニュアルページおよび『Oracle Solarisの管理:IP サービス』の「dladm コマンドによるトンネルの構成と管理」を参照してください。</td>
</tr>
<tr>
<td>[encaplimit n] [-encaplimit] [thoplimit n]</td>
<td></td>
</tr>
<tr>
<td>[auth_algs authentication algorithm] [encr_algs encryption algorithm] [encr_auth_algs encryption authenticatiion algorithm]</td>
<td>ipsecconf</td>
</tr>
<tr>
<td>[auth_revarp] [ether [address]] [index if-index] [subnet subnet-address] [broadcast broadcast-address] [token address /prefix-length]</td>
<td>同等のサブコマンドは現在使用不可。</td>
</tr>
<tr>
<td>dhcp オプション - inform、ping、release、status、drop</td>
<td></td>
</tr>
<tr>
<td>modlist] [modinsert mod_name@pos] [modremove mod_name@pos]</td>
<td>同等のサブコマンドは現在使用不可。</td>
</tr>
</tbody>
</table>

### ndd コマンドのオプションと ipadm コマンドのオプション

次の表は、ndd コマンドのオプションと、ほぼそれに対応する ipadm サブコマンドを示しています。

<table>
<thead>
<tr>
<th>ndd コマンド</th>
<th>ipadm コマンド</th>
</tr>
</thead>
<tbody>
<tr>
<td>プロパティーの取得</td>
<td></td>
</tr>
</tbody>
</table>
表9-4 neddコマンドとipadmコマンドの構文の対応 （続き）

<table>
<thead>
<tr>
<th>nedd コマンド</th>
<th>ipadm コマンド</th>
</tr>
</thead>
<tbody>
<tr>
<td>bash-3.2# nedd -get /dev/ip ?</td>
<td>bash-3.2# ipadm show-prop ip</td>
</tr>
<tr>
<td>ip_def_ttl (read and write)</td>
<td>PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE</td>
</tr>
<tr>
<td>ipv4 forwarding rw off -- off on,off</td>
<td></td>
</tr>
<tr>
<td>ipv4 ttl rw 255 -- 255 1-255</td>
<td></td>
</tr>
<tr>
<td>ipv6 forwarding rw off -- off on,off</td>
<td></td>
</tr>
<tr>
<td>ipv6 hoplimit rw 255 -- 255 1-255</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>bash-3.2# nedd -get /dev/tcp ?</td>
<td>bash-3.2# ipadm show-prop -p ecn,sack tcp</td>
</tr>
<tr>
<td>tcp_cwnd_max (read and write)</td>
<td>PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE</td>
</tr>
<tr>
<td>tcp_extra_ rw 2049 2049,4045 2049,4045 1-65535</td>
<td></td>
</tr>
<tr>
<td>tcp usehost_ rw 53843 -- 53843 3-16384</td>
<td></td>
</tr>
<tr>
<td>tcp_time_wait_interval (read and write)</td>
<td>tcp maxbuf</td>
</tr>
<tr>
<td>tcp recv_ rw 128000 -- 128000 2048-1073741824</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>bash-3.2# nedd -get /dev/tcp ecn 1</td>
<td>tcp send_ rw 49152 -- 49152 4096-1073741824</td>
</tr>
<tr>
<td>bash-3.2# nedd -get /dev/tcp sack 2</td>
<td>tcp maxbuf</td>
</tr>
<tr>
<td>tcp smallest_ rw 32768 -- 32768 1024-65535</td>
<td></td>
</tr>
<tr>
<td>tcp smallest_ nonpriv_port rw 1024 -- 1024 1024-32768</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

プロパティの設定
### 9-4 nddコマンドとipadmコマンドの構文の対応

<table>
<thead>
<tr>
<th>ndd コマンド</th>
<th>ipadmコマンド</th>
</tr>
</thead>
<tbody>
<tr>
<td>bash-3.2# ndd -set /dev/ip \ ip_def_ttl 64</td>
<td>bash-3.2# ipadm set-prop -p ttl=64 ipv4</td>
</tr>
<tr>
<td>bash-3.2# ndd -get /dev/ip \ ip_def_ttl 64</td>
<td>bash-3.2# ipadm show-prop -p ttl ip</td>
</tr>
</tbody>
</table>

PROTO PROPERTY FAMILY PERM VALUE DEFAULT POSSIBLE

<table>
<thead>
<tr>
<th>PROTO</th>
<th>PROPERTY</th>
<th>PERM</th>
<th>CURRENT</th>
<th>PERSISTENT</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv4</td>
<td>ttl</td>
<td>rw</td>
<td>64</td>
<td>64</td>
<td>255</td>
<td>1-255</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

bash-3.2# ipadm reset-prop -p ttl ip
bash-3.2# ipadm show-prop -p ttl ip

PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE

<table>
<thead>
<tr>
<th>PROTO</th>
<th>PROPERTY</th>
<th>PERM</th>
<th>CURRENT</th>
<th>PERSISTENT</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv4</td>
<td>ttl</td>
<td>rw</td>
<td>255</td>
<td>255</td>
<td>255</td>
<td>1-255</td>
</tr>
</tbody>
</table>
Oracle Solaris 上での無線インタフェース通信の構成

この章では、Oracle Solaris を実行するノートパソコン上で無線インタフェース通信を構成して使用する方法について説明します。次の項目が含まれています。

- WiFi インタフェース上の通信
- WiFi ネットワークの検索
- Oracle Solaris システム上での WiFi の接続および使用
- セキュリティー保護された WiFi 通信

WiFi 通信のタスクマップ

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>システム上の WiFi 通信を計画します。</td>
<td>WiFi をサポートする場所に、ノートパソコンまたは無線ネットワーク構成(必要に応じて、ルーターを含む)を設定します。</td>
<td>212 ページの「システムを WiFi 通信のために準備する方法」</td>
</tr>
<tr>
<td>WiFi ネットワークに接続します。</td>
<td>ローカルの WiFi ネットワークとの通信を設定して確立します。</td>
<td>212 ページの「WiFi ネットワークに接続する方法」</td>
</tr>
<tr>
<td>WiFi リンク上の通信を監視します。</td>
<td>標準の Oracle Solaris ネットワークツールを使用して WiFi リンクの状態を確認します。</td>
<td>217 ページの「WiFi リンクを監視する方法」</td>
</tr>
<tr>
<td>セキュリティー保護された WiFi 通信を確立します。</td>
<td>WEP キーを作成し、それを使 用してセキュリティ保護された WiFi ネットワークとの接続を確立します。</td>
<td>219 ページの「暗号化された WiFi ネットワーク接続を設定する方法」</td>
</tr>
</tbody>
</table>

209
WiFi インタフェース上の通信

IEEE 802.11 仕様では、ローカルエリアネットワークのための無線通信が定義されております。これらの仕様と、そこに記載されているネットワークはまとめて「WiFi」と呼ばれること。この用語は、Wi-Fi Alliance 業界グループによって商標登録されています。WiFi ネットワークは、プロバイダと想定クライアントの両方が比較的容易に構築できます。そのため、WiFi ネットワークはますます普及し、世界中で一般的に使用されています。WiFi ネットワークでは、携帯電話、テレビ、およびラジオと同じ電波テクノロジー使用されます。

Oracle Solaris には、システムを WiFi クライアントとして構成できる機能が含まれています。この節では、dtdad コマンドの WiFi 接続オプションを使用して、ノートパソコンやホームコンピュータをローカルの WiFi ネットワークに接続する方法について説明します。

注 - Oracle Solaris には、WiFi サーバーまたはアクセスポイントを構成するための機能は含まれていません。

WiFi ネットワークの検索

WiFi ネットワークには通常、次の 3 種類があります。

- 商用 WiFi ネットワーク
- 自治体 WiFi ネットワーク
- プライベート WiFi ネットワーク

WiFi のサービス対象となる場所は「ホットスポット」と呼ばれています。各ホットスポットにはアクセスポイントが含まれています。「アクセスポイント」とは、インターネットへの「有線接続 (Ethernetや DSL など) を備えたルーター」のことで、インターネット接続は通常、無線インターネットサービスプロバイダ (WISP) または従来の ISP から提供されます。

商用 WiFi ネットワーク

多くのホテルやカフェには、ノートパソコンを持った顧客へのサービスとして無線インターネット接続が提供されています。これらの商用ホットスポットは、その施設内にアクセスポイントを備えています。これらのアクセスポイントは、商用の場所にサービスを提供する WISP への有線接続を備えたルーターです。標準的な WISP には、独立系のプロバイダや携帯電話会社が含まれます。

Oracle Solaris を実行するノートパソコンを使用して、ホテルやその他の商用ホットスポットによって提供される WiFi ネットワークに接続することができます。WiFi ネットワークに接続するための手順については、各ホットスポットで尋ねてください。
WiFiインタフェース上の通信

い。通常、接続プロセスでは、ログイン時に起動するブラウザにキーを指定する必要があります。このネットワークを使用するには、ホテルまたはWISPへの料金の支払いが必要になることがあります。

インターネットホットスポットである商用の場所は通常、この機能を利用者に通知します。また、無線ホットスポットの一覧を、さまざまなWebサイト（たとえば、Wi-FiHotSpotList.com（http://www.wi-fihotspotlist.com））から検索することもできます。

自治体WiFiネットワーク

世界中の都市で、市民が自宅のシステムからアクセスできる無料の自治体WiFiネットワークが構築されてきました。自治体WiFiは、電柱やその他の屋外の場所に設置した無線送信機を使用して、ネットワークのサービス対象地域にわたる「網の目」を形成しています。これらの送信機は、自治体WiFiネットワークへのアクセスポイントです。住んでいる地域が自治体WiFiネットワークのサービス対象になっている場合は、自宅がネットワークの網の目に含まれている可能性があります。

自治体WiFiへのアクセスは通常、無料です。Oracle Solarisを実行する、正しく装備されたノートパソコンまたはパーソナルコンピュータから自治体ネットワークにアクセスできます。システムから自治体ネットワークにアクセスするためにホームルーターは必要ありません。ただし、自治体ネットワークからの信号が弱い地域では、ホームルーターを構成することをお勧めします。また、WiFiネットワーク経由のセキュリティ保護された接続が必要な場合にも、ホームルーターをお勧めします。詳細は、218ページの「セキュリティ保護されたWiFi通信」を参照してください。

プライベートWiFiネットワーク

WiFiネットワークは構成が比較的容易であるため、企業や大学では、従業員や学生にアクセスが制限されたプライベートWiFiネットワークが使用されています。プライベートWiFiネットワークでは通常、接続したあと、セキュリティ保護されたVPNを接続または実行するときにキーを指定する必要があります。セキュリティ機能を使用してプライベートネットワークに接続するには、Oracle Solarisを実行する、正しく装備されたノートパソコンまたはPCとアクセス権が必要です。

WiFi通信の計画

システムをWiFiネットワークに接続する前に、次の手順を完了してください。
システムを WiFi 通信のために準備する方法

1 サポートされている WiFi インタフェースをシステムに装備します。
システムに、Oracle Solaris でサポートされている WiFi カード (Atheros チップセットをサポートするカードなど) が装備されている必要があります。現在サポートされているドライバとチップセットの一覧については、Wireless Networking for OpenSolaris (http://hub.opensolaris.org/bin/view/Community+Group+laptop/wireless) を参照してください。

インタフェースがまだシステム上に存在しない場合は、インタフェースカードを取り付けるための製造元の指示に従ってください。インタフェースのソフトウェアは、212 ページの「WiFi インタフェースとネットワーク」の手順中で構成します。

2 システムを WiFi ネットワーク（商用、自治体、プライベートのいずれか）のサービスに対象になっている場所に設置します。
システムを、ネットワークのアクセスポイントの近くに置く必要があります。商用またはプライベートネットワークのホットスポットでは通常、この点を考慮する必要はありません。ただし、無料の自治体ネットワークの使用を計画している場合も、送信機アクセスポイントに近い場所にすることであります。

3 (省略可能) 追加のアクセスポイントとして機能する無線ルーターを設定します。
現在の場所で WiFi ネットワークが使用できない場合は、ユーザーや独自のルーターを設定します。とえば、DSL 回線がある場合は、その DSL ルーターに無線ルーターを接続します。それにより、その無線ルーターが無線デバイスのアクセスポイントになります。

Oracle Solaris システム上での WiFi の接続および使用

この節では、Oracle Solaris を実行するノートパソコンまたはデスクトップコンピュータで WiFi 接続を確立および監視するためのタスクが含まれています。

WiFi ネットワークに接続する方法

始める前に
次の手順では、212 ページの「システムを WiFi 通信のために準備する方法」の手順に従っていることを前提にしています。

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。
2 使用可能なリンクを確認します。

```
# dladm show-link
```

<table>
<thead>
<tr>
<th>LINK</th>
<th>CLASS</th>
<th>MTU</th>
<th>STATE</th>
<th>BRIDGE</th>
<th>OVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>ath0</td>
<td>phy</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>e1000g0</td>
<td>phy</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

この例の出力は、2つのリンクが使用可能なことを示しています。ath0 リンクは、WiFi 通信をサポートしています。e1000g リンクは、システムを有線ネットワークに接続するものです。

3 WiFi インタフェースを構成します。
次の手順を使用してインタフェースを構成します。

- WiFi をサポートするインタフェースを作成します。

```
# ipadm create-ip ath0
```

- リンクが plumb されていることを確認します。

```
# ipadm show-if
```

<table>
<thead>
<tr>
<th>IFNAME</th>
<th>CLASS</th>
<th>STATE</th>
<th>ACTIVE</th>
<th>OVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo0</td>
<td>loopback</td>
<td>ok</td>
<td>yes</td>
<td>--</td>
</tr>
<tr>
<td>e1000g0</td>
<td>ip</td>
<td>ok</td>
<td>yes</td>
<td>--</td>
</tr>
<tr>
<td>ath0</td>
<td>ip</td>
<td>ok</td>
<td>yes</td>
<td>--</td>
</tr>
</tbody>
</table>

4 使用可能なネットワークを確認します。

```
# dladm scan-wifi
```

<table>
<thead>
<tr>
<th>LINK</th>
<th>ESSID</th>
<th>BSSID/IBSSID</th>
<th>SEC</th>
<th>STRENGTH</th>
<th>MODE</th>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>ath0</td>
<td>net1</td>
<td>00:0e:38:49:01:d0</td>
<td>none</td>
<td>good</td>
<td>g</td>
<td>54Mb</td>
</tr>
<tr>
<td>ath0</td>
<td>net2</td>
<td>00:0e:38:49:02:f0</td>
<td>none</td>
<td>very weak</td>
<td>g</td>
<td>54Mb</td>
</tr>
<tr>
<td>ath0</td>
<td>net3</td>
<td>00:0d:ed:a5:47:e0</td>
<td>none</td>
<td>very good</td>
<td>g</td>
<td>54Mb</td>
</tr>
</tbody>
</table>

scan-wifi コマンドのこの出力例では、現在の場所で使用可能な WiFi ネットワークに関する情報が表示されています。この出力には、次の情報が含まれています。

- **LINK**
  WiFi 接続で使用されるリンク名。

- **ESSID**
  拡張サービスセット識別子。ESSID は、WiFi ネットワークの名前です。この出力例では、net1、net2、net3 などです。

- **BSSID/IBSSID**
  基本サービスセット識別子。特定の ESSID の一意の識別子です。BSSID は、ネットワークに特定の ESSID を提供する、隣接するアクセスポイントの 48 ビットの MAC アドレスです。

- **SEC**
  ネットワークにアクセスするために必要なセキュリティーのタイプ。この値は none または WEP です。WEP については、218 ページの「セキュリティー保護された WiFi 通信」を参照してください。

- **STRENGTH**
  現在の場所で使用可能な WiFi ネットワークからの無線信号の強度。
MODE
ネットワークによって実行される 802.11 プロトコルのバージョン。このモードは a、b、g か、またはこれらのモードの組み合わせです。

SPEED
特定のネットワークの速度（メガビット/秒）。

5 WiFiネットワークに接続します。
次のいずれかを行います。

- 信号強度のもっとも高い、セキュリティーで保護されていない WiFi ネットワークに接続します。

  # dladm connect-wifi

- ESSID を指定することによって、セキュリティーで保護されていないネットワークに接続します。

  # dladm connect-wifi -e ESSID

dladm の connect-wifi サブコマンドには、WiFi ネットワークに接続するためのその他のオプションがいくつかあります。詳細は、dladm(1M) のマニュアルページを参照してください。

6 インタフェースのIPアドレスを構成します。
次のいずれかを行います。

- DHCP サーバーから IP アドレスを取得します。

  # ipadm create-addr -T dhcp addrobj

  ここで、addrobj は interface/user-defined-string の命名規則を使用します。

  WiFi ネットワークで DHCP がサポートされていない場合は、次のメッセージが表示されます。

  ipadm: interface: interface does not exist or cannot be managed using DHCP

- 静的 IP アドレスを構成します。

  このオプションは、システムに専用の IP アドレスがある場合に使用します。

    # ipadm create-addr -T static -a address addrobj

7 システムの接続先のWiFiネットワークのステータスを確認します。

<table>
<thead>
<tr>
<th>LINK</th>
<th>STATUS</th>
<th>ESSID</th>
<th>SEC</th>
<th>STRENGTH</th>
<th>MODE</th>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>ath0</td>
<td>connected</td>
<td>net3</td>
<td>none</td>
<td>very good</td>
<td>g</td>
<td>36Mb</td>
</tr>
</tbody>
</table>

この例の出力は、システムが現在 net3 ネットワークに接続されていることを示しています。前の scan-wifi の出力は、使用可能なネットワークの中で net3 の信号強度
8 WiFi ネットワーク経由でインターネットにアクセスします。
システムの接続先のネットワークに応じて、次のいずれかを行います。

- アクセスポイントが無料のサービスを提供している場合は、今すぐ目的のブラウザまたはアプリケーションを実行できます。
- アクセスポイントが料金を必要とする商用ホットスポット内にある場合は、現在の場所で提供されている手順に従います。通常は、ブラウザを実行し、キーを指定して、ネットワークプロバイダにクレジットカード情報を提供します。

9 セッションを完了します。
次のいずれかを行います。

- WiFi セッションを終了しますが、システムは実行中のままにします。
  
  ```
  # dladm disconnect-wifi
  ```

- 現在、複数のセッションが実行されている場合は、特定の WiFi セッションを終了します。
  
  ```
  # dladm disconnect-wifi link
  ```
  ここで、link はセッションに使用されていたインタフェースを表します。

- WiFi セッションの実行中に、システムを正常にシャットダウンします。

  ```
  # shutdown -g0 -i5
  ```

shutdown コマンドを使用してシステムの電源を落とす前に WiFi セッションを明示的に切り離す必要はありません。

例 10-1 特定の WiFi ネットワークへの接続
次の例は、インターネットカフェで、Oracle Solaris を実行するノートパソコンを使用しているときに発生する可能性のある標準的なシナリオを示しています。

WiFi リンクが使用可能かどうかを調べます。

```
# dladm show-wifi
ath0 type: non-vlan mtu: 1500 device: ath0
```

ath0 リンクは、ノートパソコン上にインストールされています。ath0 インタフェースを構成し、それが稼働していることを確認します。

```
# ipadm create-ip ath0
IFNAME STATE CURRENT PERSISTENT
```
現在の場所で使用可能な WiFi リンクを表示します。

```
# dladm scan-wifi
LINK ESSID BSSID/IBSSID SEC STRENGTH MODE SPEED
ath0 net1 00:0e:38:49:01:d0 none weak g 54Mb
ath0 net2 00:0e:38:49:02:f0 none very weak g 54Mb
ath0 net3 00:0d:ed:a5:47:e0 wep very good g 54Mb
ath0 citinet 00:40:96:2a:56:b5 none good b 11Mb
```

この出力は、net3 の信号が最適であることを示します。net3 にはキーが必要です。このキーに対して、カフェのプロバイダは料金を請求します。citinet は、その土地で提供されている無料のネットワークです。

```
citinet ネットワークに接続します。
```

```
# dladm connect-wifi -e citinet
connect-wifi の -e オプションには、優先 WiFi ネットワークの ESSID を引数として指定します。このコマンドでの引数は、無料のローカルネットワークの ESSID である citinet です。dladm connect-wifi コマンドでは、WiFi ネットワークに接続するためのオプションがいくつか提供されています。詳細は、dladm(1M) のマニュアルページを参照してください。
```

WiFi インタフェースの IP アドレスを構成します。

```
# ipadm create-addr -T static -a 10.192.16.3/8 ath0/v4
# ipadm show-addr
ADDROBJ TYPE STATE ADDR
lo0/v4 static ok 127.0.0.1/8
e1000g0/v4 static ok 129.146.69.34/24
ath0/v4static static ok 10.192.16.3/8
lo0/v6 static ok ::1/128
```

この例では、ノートパソコン上に静的 IP アドレス 10.192.16.3/24 が構成されていることを前提にしています。

```
# dladm show-wifi
LINK STATUS ESSID SEC STRENGTH MODE SPEED
ath0 connected citinet none good g 11Mb
```

この出力は、このノートパソコンが現在、ネットワーク citinet に接続されていることを示します。

```
# firefox
Firefox ブラウザのホームページが表示されます。
```
ブラウザやその他のアプリケーションを実行して、WiFiネットワーク経由で作業を開始します。

```
# dladm disconnect-wifi
# dladm show-wifi
LINK   STATUS   ESSID   SEC   STRENGTH MODE   SPEED
ath0   disconnected --   --   --   --   --   --
```

`show-wifi`の出力により、WiFiネットワークからath0リンクを切り離していることが確認されます。

▼ WiFiリンクを監視する方法

この手順は、標準のネットワークツールを使用してWiFiリンクのステータスを監視したり、linkpropサブコマンドを使用してリンクプロパティを変更したりする方法を示しています。

1 管理者になります。
詳細は、『Oracle Solarisの管理:セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 212ページの「WiFiネットワークに接続する方法」の説明に従って、WiFiネットワークに接続します。

3 リンクのプロパティを表示します。
構文は次のとおりです。
```
# dladm show-linkprop interface
```
たとえば、ath0リンクを通して確立された接続のステータスを表示するには、次の構文を使用します。

```
# dladm show-linkprop ath0
PROPERTY VALUE DEFAULT POSSIBLE
channel 5   --   --   --   --   --   --   --
powermode off off off,fast,max
radio ?    on    on,off
speed 36   --   1,2,5.5,6,9,11,12,18,24,36,48,54
```

4 リンクの固定された速度を設定します。

注意 - Oracle Solarisでは、WiFi接続の最適な速度が自動的に選択されます。初期のリンク速度を変更すると、パフォーマンスが低下したり、特定のWiFi接続の確立が妨げられたりすることもあります。
センターを、show-linkprop の出力に表示される、速度に指定可能な値のいずれかに変更できます。

```
# dladm set-linkprop -p speed=value link
```

5 リンク上のパケットフローを確認します。

```
# netstat -I ath0 -i 5
```

<table>
<thead>
<tr>
<th>input</th>
<th>ath0</th>
<th>output</th>
<th>colls</th>
<th>input (Total)</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>packets</td>
<td>err</td>
<td>packets</td>
<td>err</td>
<td>packets</td>
<td>err</td>
</tr>
<tr>
<td>317</td>
<td>0</td>
<td>106</td>
<td>0</td>
<td>0</td>
<td>2905</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>304</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>631</td>
</tr>
<tr>
<td>338</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>722</td>
</tr>
<tr>
<td>294</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>670</td>
</tr>
<tr>
<td>306</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>649</td>
</tr>
<tr>
<td>289</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>597</td>
</tr>
</tbody>
</table>

例 10-2 リンクの速度の設定

この例は、WiFi ネットワークに接続したあとにリンクの速度を設定する方法を示しています。

```
# dladm show-linkprop -p speed ath0
PROPERTY VALUE DEFAULT POSSIBLE
speed 24 -- 1,2,5,6,9,11,12,18,24,36,48,54
```

```
# dladm set-linkprop -p speed=36 ath0
```

```
# dladm show-linkprop -p speed ath0
PROPERTY VALUE DEFAULT POSSIBLE
speed 36 -- 1,2,5,6,9,11,12,18,24,36,48,54
```

セキュリティー保護された WiFi 通信

電波テクノロジーによって WiFi ネットワークが簡単に使用できるように、さらにたい
ては、多くの場所にいるユーザーから自由にアクセスできるようになっている。
その結果、WiFi ネットワークへの接続が安全でなくなる場合があります。ただし、特定のタイプの WiFi 接続は、よりセキュリティーが強化されています。

1 アクセスが制限されたプライベート WiFi ネットワークへの接続

企業や大学によって確立された内部ネットワークなどのプライベートネット
ワークは、各ネットワークへのアクセスを、正しいセキュリティーチェックを
指定できるユーザーに制限しています。潜在的なユーザーは、接続シーンス中
にキーを指定するか、またはセキュリティー保護された VPN を経由してネット
ワークにログインする必要があります。

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
■ WiFi ネットワークへの接続の暗号化

セキュリティ保護されたキーを使用して、システムとWiFiネットワークの間の通信を暗号化できます。WiFiネットワークへのアクセスポイントは、セキュリティ保護されたキーを生成する機能を備え、自宅またはオフィスのルーターである必要があります。システムとルーターは、セキュリティ保護された接続を作成するためにキーを確立してから共有します。

dladmコマンドでは、アクセスポイントを経由した接続を暗号化するためのWEP (Wired Equivalent Privacy)キーを使用できます。WEPプロトコルは、無線接続のためのIEEE 802.11仕様で定義されています。dladmコマンドのWEP関連のオプションについての詳細は、dladm(1M)のマニュアルページを参照してください。

▼ 暗号化されたWiFiネットワーク接続を設定する方法

次の手順は、システムと自宅のルーターの間のセキュリティ保護された通信を設定する方法を示しています。家庭向けの無線および有線ルーターの多くには、セキュリティ保護されたキーを生成できる暗号化機能があります。この手順では、このようなルーターを使用すること、およびそのマニュアルが参照可能であることを前提にしています。また、システムがすでにルーターに接続されていることとも前提にしています。

1 ホームルーターを構成するためのソフトウェアを起動します。
手順については、製造元のマニュアルを参照してください。ルーターの製造元は通常、ルーター構成のための内部のWebサイトまたはグラフィカルユーザーアイテュースを提供しています。

2 WEPキーの値を生成します。
ルーターのセキュリティ保護されたキーを作成するための製造元の指示に従ってください。ルーター構成のGUIから、キーの希望するパスフレーズを指定するよう求められることもあります。ソフトウェアに次に、このパスフレーズを使用して16進文字列(通常は、5バイトまたは13バイトの長さ)を作成します。この文字列が、WEPキーに使用される値になります。

3 キーの構成を適用して保存します。
手順については、製造元のマニュアルを参照してください。

4 管理者になります。
詳細は、『Oracle Solarisの管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。
セキュリティー保護されたWiFi通信

5 WEPキーを含むセキュアオブジェクトを作成します。
システム上に端末ウィンドウを開き、次を入力します。

```bash
# dladm create-secobj -c wep keyname
```
ここで、`keyname`はキーに付ける名前を表します。

6 セキュアオブジェクトにWEPキーの値を指定します。
`create-secobj`サブコマンドは次に、キーの値を要求するスクリプトを実行します。

```bash
provide value for keyname: 5 or 13 byte key
confirm value for keyname: retype key
```
この値は、ルーターによって生成されたキーです。スクリプトは、キーの値として5バイトまたは13バイトの文字列(ASCIIまたは16進数)のどちらかを受け入れます。

7 今作成したキーの内容を表示します。

```bash
# dladm show-secobj
OBJECT CLASS
keyname wep
```
ここで、`keyname`は、このセキュアオブジェクトの名前です。

8 WiFiネットワークへの暗号化された接続を作成します。

```bash
# dladm connect-wifi -e network -k keyname interface
```

9 接続がセキュリティーで保護されていることを確認します。

```bash
# dladm show-wifi
LINK STATUS ESSID SEC STRENGTH MODE SPEED
ath0 connected net1 wep good g 11Mb
```
SECの見出しの下の`wep`の値は、この接続にWEP暗号化が適用されていることを示します。

例 10-3 暗号化されたWiFi通信の設定
この例では、次のことがすでに実行されていることを前提にしています。

- システムを、WEPキーを作成できるホームルーターに接続します。
- ルーターの製造元のマニュアルに従ってWEPキーを作成します。
- キーを保存することにより、それを使用してシステム上にセキュアオブジェクトを作成できるようにします。

```bash
# dladm create-secobj -c wep mykey
provide value for mykey: *****
confirm value for mkey: *****
```
ルーターによって生成されたWEPキー指定するときは、入力した値がアスタリスクでマスクされます。

Oracle Solaris管理: ネットワークインターフェースとネットワーク仮想化・2011年12月 220
セキュリティー保護されたWiFi通信

```
# dladm show-secobj
OBJECT      CLASS
    mykey      wep

# dladm connect-wifi -e citinet -k mykey ath0

このコマンドは、セキュアオブジェクトmykeyを使用して、WiFiネットワークcitinetへの暗号化された接続を確立します。

# dladm show-wifi
LINK   STATUS   ESSID   SEC   STRENGTH MODE  SPEED
ath0   connected citinet wep good g 36Mb
```

この出力により、WEP暗号化を使用してcitinetに接続されていることが確認されます。
第11章

ブリッジを管理する

この章では、ブリッジおよびその管理方法について説明します。

この章で扱う内容は、次のとおりです。

- 223ページの「ブリッジングの概要」
- 234ページの「ブリッジの管理（タスクマップ）」

ブリッジングの概要

ブリッジは、別々のネットワークセグメントを接続する際に使用されます。ブリッジで接続すると、接続されたネットワークセグメントは単一のネットワークセグメントであるかのように通信を行います。ブリッジングは、ネットワークスタックのデータリンク層 (L2) で実装されます。ブリッジはパケット転送メカニズムを使用して、複数のサブネットワークを相互に接続します。

ブリッジングとルーティングはどちらも、ネットワーク上のリソースの場所に関する情報を配布するために使用できますが、いくつかの点で異なります。ルーティングはIP層 (L3) で実装され、ルーティングプロトコルを使用します。データリンク層では、ルーティングプロトコルは使用されません。代わりに、パケットの転送先は、ブリッジに接続されているリンクで受信されたネットワークトライフィックを検査することで特定されます。

パケットを受信すると、その発信元アドレスが検査されます。パケットの発信元アドレスには、受信されたリンクにパケットを送信したノードが関連付けられますが。その後、受信されたパケットが宛先アドレスと同じアドレスを使用すると、ブリッジはリンク上のパケットをそのアドレスに転送します。

発信元アドレスと関連付けられたリンクは、ブリッジサブネットワーク内の別のブリッジに接続された中間リンクである場合があります。時間が経過すると、ブリッジサブネットワーク内のすべてのブリッジが、どのリンクが特定のノードにパ
プリッジの概要

ケットを送信するのかを「学習」します。したがって、トップバイトップのプリッジ方式で最終宛先にパケットを送信する際には、パケットの宛先アドレスが使用されます。

ローカルの「リンク停止」通知は、特定のリンク上のすべてのノードが到達不能になったことを示します。この状況では、リンクへのパケット転送が停止し、リンク上のすべての転送エンジンがフラッシュされます。時間が経過すると、転送エンジンも消去されます。リンクが復元されると、リンク上で受信されたパケットは廃棄として処理されます。パケットの発送元アドレスに基づいた「学習」プロセスが再開します。このプロセスによって、アドレスが宛先アドレスとして使用されたときに、プリッジが正常にパケットをそのリンク上に転送できるようになります。

パケットを宛先に転送するには、プリッジに接続されたすべてのリンク上でプリッジがプロミスキャスモードで待機する必要があります。プロミスキャスモードで待機すると、最大回線速度でパケットを永久に循環してしまう転送ループの発生に対して、プリッジが脆弱になります。したがって、プリッジではスパニングツリープロトコル (STP) メカニズムを使用して、サブネットワークが使用不可になるネットワークループを回避します。

プリッジに STP およびラピッドスパニングツリープロトコル (RSTP) を使用することに加えて、Oracle Solaris では TRILL 保護拡張機能もサポートされています。デフォルトでは STP が使用されますが、プリッジングコマンドで -P trill オプションを指定すれば、TRILL も使用できます。

プリッジ構成を使用して、単一のネットワークに接続すると、ネットワーク内のさまざまなノードの管理が簡略化されます。これらのセグメントをプリッジ経由で接続すると、すべてのノードが単一のブロードキャストネットワークを共有します。したがって、ルーターを使用してトラフィックをネットワークセグメント間で転送するのではなく、IP などのネットワークプロトコルを使用することによって、各ノードは他のノードに到達できます。プリッジを使用しない場合、ノード間の IP トラフィックの転送を許可するように IP ルーティングを構成する必要があります。

次の図は、単純なプリッジネットワーク構成を示しています。プリッジ goldenigate は、プリッジングが構成された Oracle Solaris システムです。sanfrancisco および sausalito は、物理的にプリッジに接続されたシステムです。ネットワーク A ではハブが使用され、片側では物理的にプリッジに接続され、反対側ではコンピュータシステムに接続されています。プリッジポートは、bge0、bge1、bge2 などのリンクです。
ブリッジングの概要

図11-1 単純なブリッジネットワーク

ブリッジネットワークはリング状に形成して、物理的に複数のブリッジを相互に接続できます。このような構成は、ネットワークでよく使用されます。このタイプの構成では、古いパケットがリングの周りをエンドレスにループすることによってネットワークリンクが飽和状態になる問題が発生する可能性があります。このようなループ状態を保護するために、Oracle Solaris のブリッジには STP と TRILL の両方のプロトコルが実装されています。大部分のハードウェアブリッジでは、STP ループ保護も実装されています。

次の図は、リング状に構成されたブリッジネットワークを示しています。この構成では、3 台のブリッジがあります。2 台のシステムが物理的に westminster に接続されています。1 台のシステムが物理的に waterloo に接続されています。さらに、1 台のシステムが物理的に tower に接続されています。各ブリッジは、物理的にブリッジポートを介して相互に接続されています。

ループ保護のために STP または RSTP を使用すると、ループ内の接続の 1 つによるパケット転送を回避することによって、物理的なループが軽減されます。この図は、westminster と tower ブリッジ間の物理リンクがパケット転送に使用されないことを示しています。

ループ保護を実行するために使用可能な物理リンクをシャットダウンすると、STP および RSTP によって帯域幅が消費されることに注意してください。

STP および RSTP とは異なり、TRILL ではループを回避するために物理リンクがシャットダウンされません。その代わりに、TRILL はネットワーク内の TRILL ノードごとに最短パス情報を計算し、その情報を使用して個々の宛先にパケットを転送します。

その結果、TRILL を使用すると、すべてのリンクが常に使用中のまま保持できます。IP によるループの処理方法と同様にループが処理されるため、ループが問題に
あることはありません。つまり、TRILLは必要に応じてルートを作成し、転送のホップ制限を使用することで、一時的なループ状態によって発生する問題を回避します。

図11-2 ブリッジネットワークリング

注意 - SPARCプラットフォームでは、local-mac-address?=falseを設定しないでください。これを設定すると、複数のポートや同じネットワーク上で同一のMACアドレスが重複して使用されます。

注 - 可能な限り高いレベルのパフォーマンスが必要な場合は、ブリッジリンクを構成しないでください。ブリッジでは、ベースとなるインタフェースをプロミスキャスモードにして、システムのハードウェア層、ドライバ層、およびその他の層で実装されている数多くの重要な最適化機能を無効にする必要があります。このようなパフォーマンス拡張機能が無効になることは、ブリッジングメカニズムでは避けられない結果です。

システムのリンクの一部がブリッジされていないために、これらの制約の対象ではないシステムでは、ブリッジを使用できます。このようなパフォーマンスの問題は、ブリッジの一部になるように構成されたリンクにのみ影響を与えます。
STPの詳細は、IEEE 802.1D-1998を参照してください。RSTPの詳細は、IEEE 802.1Q-2004を参照してください。TRILLの詳細は、Internet Engineering Task Force (IETF) TRILL draft documents (http://tools.ietf.org/wg/trill)を参照してください。

リンクプロパティー

次のリンクプロパティーは、dladm show-linkprop、dladm set-linkprop、および resets-linkpropコマンドで表示および変更できます。

default_tag  リンク間で送受信されるタグなしパケットに、デフォルトの仮想ローカルエリアネットワーク (VLAN) IDを定義します。有効な値は0-4094です。デフォルト値は1です。このプロパティは、VLAN以外および仮想ネットワークインタフェースカード (VNIC) 以外のタイプのリンクにのみ存在します。この値を0に設定すると、ポートとの間で送受信されるタグなしパケットの転送が無効になります (これはMACプロパティーです)。

注 - このプロパティーは、デフォルトの仮想ポートVLAN Identifier (PVID)を指定することもできます。ベースリンク自体が自動的にPVIDを表すため、default_tagをゼロ以外にすると、リンク上に同じIDを持つVLANを作成できません。

たとえば、net0でPVIDを5に設定する場合は、net0でID5のVLANを作成できません。この状況でVLAN5を指定するには、net0を使用します。

default_tagは、そのリンクで作成された既存のVLANのIDと等しくなるように設定できません。たとえば、次のコマンドはnet0にVLAN22を作成します。

```
# dladm create-vlan -l net0 -v 22 myvlan0
```

この状況では、net0とmyvlan0の両方で同じVLANを表すため、default_tagを22に設定できません。

default_tagを0に設定すると、net0上のタグなしパケットとVLANとのあらゆる関連付けを解除できます。この状況では、このようなパケットが構成されたブリッジから転送されなくなります。

forward ブリッジ経由のトラフィック転送を有効および無効にします。このプロパティーは、VNICリンク以外のすべてのリンクに存在します。
ブリッジングの概要

STP と RSTP を有効および無効にします。有効な値は 1(真) と 0(偽) です。デフォルト値は 1 です。無効にすると、リンクインスタンスに関連付けられた VLAN では、トラフィックがブリッジから転送されなくなります。転送を無効にすることは、従来のブリッジで「許可された設定」から VLAN を削除することに相当します。つまり、ローカルクライアントからベースとなるリンクへの VLAN ベースの入出力は続行されますが、ブリッジベースの転送は実行されません。

stp

リンクを使用する際の STP と RSTP のコスト値を表示します。有効な値は 1-65535 です。デフォルト値は 0 であり、自動的にコストがリンクタイプ別に計算されるように指定されます。次の値は、複数のリンクタイプのコストを表します: 10 Mbps では 100, 100 Mbps では 19, 1 Gbps では 4, 10 Gbps では 2。

stp_cost

ポートがほかのブリッジに接続されているかどうかを指定します。有効な値は 1(真) と 0(偽) です。デフォルト値は 1 です。0 に設定されていると、どのタイプの BPDU も認識されない場合でも、デーモンはポートがほかのブリッジに接続されているとみなします。

stp_edge

接続モードタイプを指定します。有効な値は true, false, および auto です。デフォルト値は auto であり、ポイントツーポイント接続が自動的に検出されます。true を指定すると、ポイントツーポイントモードが強制され、false を指定すると、標準のマルチポイントモードが強制されます。

stp_p2p

STP および RSTP ポートの優先度値を設定します。有効な値は 0-255 です。デフォルト値は 128 です。STP および RSTP ポートの優先度値をポート識別子の先頭に付加することで、ブリッジの優先ルートポートが決定されます。数値を小さくすると、優先度が高くなります。
STP デーモン

dladm create-bridge コマンドを使用して作成された各ブリッジは、svc:/network/bridge の同じ名前の付いたSMFインスタンスとして表されます。各インスタンスは、STP を実装する /usr/lib/bridged デーモンのコピーを実行します。

次のコマンド例では、pontevecchio というブリッジを作成します。

```
# dladm create-bridge pontevecchio
svc:/network/bridge:pontevecchio という SMF サービス、および /dev/net pontevecchio0 という可観測性ノードが作成されます。
```

安全のため、デフォルトで標準 STP がすべてのポートで実行されます。STP などの一部のブリッジプロトコル形式が実行されないブリッジでは、ネットワークで長期にわたる転送ループが発生する可能性があります。Ethernet のパケットはホップ数または TTL が存在しないため、このようなループはネットワークにとって致命的です。

特定のポートが別のブリッジに接続されていない（たとえば、ホストシステムに直接ポイントツーポイント接続されている）ことが判明したときは、管理上そのポートの STP を無効にすることができます。ブリッジ上のすべてのポートで STP が無効になっている場合でも、引き続き STP デーモンは実行されます。デーモンが引き続き実行される理由は、次のとおりです。

- 新たに追加されるポートを処理するため
- BPDU ガードを実装するため
- 必要に応じて、ポート上の転送を有効または無効にするため

ポートで STP が無効になっている場合、bridged デーモンは BPDU (BPDU ガード) の待機を続けます。デーモンは syslog を使用してエラーフラグを付け、ポート上の転送を無効にして、重大なネットワークの構成ミスを示します。リンクステータスが停止してから再起動すると、またはリンクを手動で削除してから再度追加すると、再びリンクが有効になります。

ブリッジの SMF サービスインスタンスを無効にするとき、STP デーモンが停止したときに、そのポート上のブリッジ転送も停止します。インスタンスが再起動すると、STP は初期状態から起動します。

TRILL デーモン

dladm create-bridge -P trill コマンドを使用して作成された各ブリッジは、svc:/network/bridge および svc:/network/routing/trill の同じ名前の付いた
SMFインスタンスとして表されます。svc:/network/routing/trillの各インスタンスでは、TRILLプロトコルを実装する/usr/lib trilldデーモンのコピーが実行されます。

次のコマンド例では、bridgeofsighsというブリッジを作成します。

```
# dladm create-bridge -P trill bridgeofsighs
```

svc:/network/bridge:bridgeofsighs と svc:/network/routing/trill:bridgeofsighsという2つのSMFサービスが作成されます。さらに、/dev/net/bridgeofsighs0という可観測性ノードも作成されます。

**ブリッジをデバッグする**

各ブリッジインスタンスには、「可観測性ノード」が割り当てられます。このノードは、/dev/net/ディレクトリに表示され、ブリッジ名の末尾に0を付加した名前が付けられます。

可観測性ノードは、snoopおよびwiresharkユーティリティで使用することを目的としています。このノードは、パケット転送が暗黙的に行われる点を除いて、標準のEthernetインタフェースと同様に動作します。可観測性ノード上のIPはplumbすることができず、パッシブオプションを使用しない限り、バインド要求(DL_BIND_REQ)も実行できません。

可観測性ノードが使用されると、ユーザが使用可能なブリッジで処理されるすべてのパケットの未変更コピーが1つ作成されます。この動作は、従来のブリッジの「監視」ポートと同様であり、通常のDLPIの「プロミスキャスモード」規則の対象になります。pfmod、またはsnoopとwiresharkユーティリティの機能を使用すると、VLAN IDに基づいてフィルタリングできます。

配信されたパケットは、ブリッジで受信されたデータを表します。

注意- ブリッジングプロセスでVLANタグが追加、削除、または変更される場合、表示されるデータには、このプロセスが発生する前の状態が記述されます。さまざまなリンクで別々のdefault_tag値が使用されると、このようなままの状況で混乱する可能性があります。

特定のリンクで転送および受信されるパケットを(ブリッジングプロセスの完了後に)確認するには、可観測性ノード上ではなく、個々のリンクでsnoopコマンドを実行します。

可観測性ノードの詳細は、354ページの「ネットワーク仮想化およびリソース制御のための可観測性機能」を参照してください。
その他のブリッジ動作

次の節では、構成でブリッジが使用された場合におけるリンク動作がどのように変化するかについて説明します。

標準のリンク動作の詳細は、253ページの「仮想ルーカルエリアネットワークの管理」を参照してください。

DLPIの動作

次に、ブリッジが有効になっているときのリンク動作の相違点について説明します。

- リンク起動 (DL_NOTE_LINK_UP) およびリンク停止 (DL_NOTE_LINK_DOWN) の通知は集約して配信されます。つまり、すべての外部リンクがリンク停止ステータスを示しているときは、MAC層を使用している上位のクライアントにもリンク停止イベントが表示されます。ブリッジ上の任意の外部リンクがリンク起動ステータスを示しているときは、すべての上位クライアントにリンク起動が表示されます。

この集約されたリンク起動およびリンク停止の報告が実行される理由は、次のとおりです。

- リンク停止が表示されると、リンク上のノードは到達不能になります。ブリッジングコードにより別のリンク経路で引き続きパケットを送受信できる場合は、これに該当しません。実際のリンクステータスを必要とする管理アプリケーションは、既存のMAC層のカーネル統計を使用してステータスを表示できます。このようなアプリケーションは、ハードウェアのステータス情報を報告し、転送には関与しないという点で、IPなどの通常のクライアントとは異なります。

- すべての外部リンクが停止すると、ブリッジ自体がシャットダウンしたかのようにステータスが表示されます。このような特殊なケースでは、何も到達できない可能性があるとシステムで認識されます。すべてのインタフェースが「実体」であり（仮想ではなく）、すべてが接続解除されている場合は、ローカルのみの通信を許可する際にブリッジを使用できないというトレードオフがあります。

- リンク固有の機能はすべて廃用になっています。実際の出力リンクは完全にクライアントによって決定されるわけではないため、特殊なハードウェアの高速化機能がサポートされているリンクでは、これらの機能を使用できません。ブリッジ転送機能は、宛先MACアドレスに基づいて出力リンクを選択する必要があります。その出力リンクはブリッジ上の任意のリンクにすることができます。

VLANの管理

デフォルトでは、システムに構成されたVLANは、ブリッジインスタンス上のすべてのポート間で転送されます。ベースとなるリンクがブリッジの一部である場合
ブリッジの概要

に、dladm create-vlan または dladm create-vnic -v コマンドを呼び出すと、該当ブリッジリンク上に指定された VLAN の転送も有効になります。

リンク上に VLAN を構成して、ブリッジ上のその他のリンク間の転送を無効にするには、dladm set-linkprop コマンドで forward プロバティーを設定して、転送を無効にする必要があります。

ベースとなるリンクがブリッジの一部として構成されているときに、ブリッジングで VLAN を自動的に有効にするには、dladm create-vlan コマンドを使用します。

標準拠の STP では、VLAN は無視されます。ブリッジングプロトコルは、タグなし BPDU メッセージを使用して 1 つのループなしのトポロジーのためを計算し、このツリーを使用してリンクを有効および無効にします。ネットワークでプロビジョニングされている複製リンクは、STP によって自動的に無効になっても、構成された VLAN が接続解除されないように構成する必要があります。つまり、ブリッジされてパックポートのあらゆるところではすべての VLAN を実行するか、すべての冗長リンクを慎重に検査する必要があります。

TRILL は、複雑な STP の規則に従う必要はありません。その代わりに、TRILL は自動的に VLAN タグが変更されていないパケットをカプセル化して、ネットワーク経由で渡します。つまり、TRILL は、単一のブリッジネットワーク内で同じ VLAN ID が再利用されている場合、分離された VLAN を結び付けます。

これは、4094 の制限よりも多くの VLAN セットを管理するために、ネットワークの分離されたセクションで VLAN タグを再利用することがある STP との重要な相違点です。このようにネットワークを管理する際には TRILL を使用できませんが、プロバイダベースの VLAN などのその他のソリューションは実装できる場合があります。

VLAN を使用した STP ネットワークでは、STP によって「不正な」リンクが無効になったときに、VLAN パーティション分割を回避するようにフェイルオーバーの特性を構成することが困難になる可能性があります。分離された VLAN で失われる比較的小さい機能性は、TRILL モデルの堅牢性で十分に補うことができます。
VLANの動作

ブリッジは許可されたVLANセットおよびdefault_tagプロバティーをリンクごとに検査することで、転送を実行します。一般的なプロセスは次のとおりです。

- 入力VLANの決定。このタスクは、リンクでパケットが受信されると開始されます。パケットが受信されると、VLANタグがチェックされます。タグが存在しない場合や、タグが優先度のみ(タグがゼロ)の場合は、該当するリンクに構成されたdefault_tag(ゼロに設定されていない場合)が内部VLANタグとみなされます。タグが存在しない、またはゼロで、default_tagがゼロの場合は、パケットが無視されます。タグなしの転送は実行されません。タグが存在し、default_tagに等しい場合も、パケットが無視されます。それ以外の場合は、入力タグが入力VLANとみなされます。

- リンクメンバーシップのチェック。入力VLANがこのリンクで許可されたVLANとして構成されていない場合は、パケットが無視されます。その後、転送が計算され、出力リンクにも同じチェックが行われます。

- タグの更新。出力リンクのVLAN(この時点でゼロ以外)がdefault_tagに等しい場合は、優先度に関係なく、パケット上のタグ(存在する場合)が削除されます。出力リンクのVLANがdefault_tagに等しくない場合は、タグが現在存在しなければ追加され、そのタグが出力パケットに設定され、現在の優先度がパケットにコピーされます。

注：転送で複数のインタフェースに送信される場合（ブロードキャスト、マルチキャスト、および不明な宛先の場合）は、出力リンクごとに個別に出力リンクのチェックおよびタグの更新を行う必要があります。転送はタグ付きの場合も、タグなしの場合もあります。

ブリッジの構成例

次の例は、ブリッジ構成およびブリッジングサービスに関する情報を表示する方法を示しています。

次のコマンドを実行すると、ブリッジに関する情報を取得できます。

```bash
# dladm show-bridge
```

<table>
<thead>
<tr>
<th>BRIDGE</th>
<th>PROTECT ADDRESS</th>
<th>PRIORITY DESROOT</th>
</tr>
</thead>
</table>

次のコマンドを実行すると、ブリッジに関するTRILLニックネーム情報を取得できます。

```bash
# dladm show-bridge -t tonowhere
```

<table>
<thead>
<tr>
<th>NICK FLAGS LINK</th>
<th>NEXTHOP</th>
</tr>
</thead>
</table>
# ブリッジの管理（タスクマップ）

Oracle Solaris では、dladm コマンドおよび SMF 機能を使用してブリッジを管理します。インスタンスの障害管理リソース識別子 (FMRI) svc:/network/bridge を使用してブリッジインスタンスを有効化、無効化、および監視するには、SMF コマンドを使用します。ブリッジを作成または削除することに加えて、ブリッジにリンクを割り当てたり、ブリッジからリンクを削除したりするには、dladm を使用します。

次の表は、ブリッジを管理するために使用可能なタスクを示しています。

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>構成されているブリッジに関する情報を表示します。</td>
<td>システムに構成されているブリッジに関する情報を表示するには、dladm show-bridge コマンドを使用します。構成されているブリッジ、リンク、統計情報、およびカーネル転送エントリに関する情報を表示します。</td>
<td>235 ページの「構成されているブリッジに関する情報を表示する方法」</td>
</tr>
<tr>
<td>ブリッジに接続されているリンクに関する構成情報を表示します。</td>
<td>システムに構成されているリンクに関する情報を表示するには、dladm show-link コマンドを使用します。リンクがブリッジに関連付けられている場合は、BRIDGE フィールドの出力を参照してください。</td>
<td>237 ページの「ブリッジリンクに関する構成情報を表示する方法」</td>
</tr>
<tr>
<td>ブリッジを作成します。</td>
<td>ブリッジを作成して、省略可能なリンクを追加するには、dladm create-bridge コマンドを使用します。デフォルトでは、ブリッジは STP を使用して作成されます。代わりに、TRILL を使用してブリッジを作成するには、-ptrill を dladm create-bridge コマンド行に追加するか、dladm modify-bridge コマンドを使用して TRILL を有効にします。</td>
<td>237 ページの「ブリッジを作成する方法」</td>
</tr>
<tr>
<td>タスク</td>
<td>説明</td>
<td>手順</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>プリッジの保護タイプを変更します。</td>
<td>プリッジの保護タイプを変更するには、dladm modify-bridge コマンドを使用します。デフォルトでは、プリッジはSTPを使用して作成されます。代わりに、TRILLを使用してプリッジを作成するには、dladm modify-bridge コマンドで -P trill を使用してTRILLを有効にします。</td>
<td>239 ページの「プリッジの保護タイプを変更する方法」</td>
</tr>
<tr>
<td>プリッジにリンクを追加します。</td>
<td>既存のプリッジに1つ以上のリンクを追加するには、dladm add-bridge コマンドを使用します。</td>
<td>239 ページの「既存のプリッジに1つ以上のリンクを追加する方法」</td>
</tr>
<tr>
<td>プリッジからリンクを削除します。</td>
<td>プリッジからリンクを削除するには、dladm remove-bridge コマンドを使用します。すべてのリンクが削除されるまで、プリッジを削除できません。</td>
<td>240 ページの「プリッジからリンクを削除する方法」</td>
</tr>
<tr>
<td>システムからプリッジを削除します。</td>
<td>システムからプリッジを削除するには、dladm delete-bridge コマンドを使用します。</td>
<td>240 ページの「システムからプリッジを削除する方法」</td>
</tr>
</tbody>
</table>

▼ 構成されているプリッジに関する情報を表示する方法

この手順では、さまざまなオプションとともに dladm show-bridge コマンドを使用して、構成されているプリッジに関する各種情報を表示する方法を示します。

dladm show-bridge コマンドオプションの詳細は、dladm(1M) のマニュアルページを参照してください。

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 1 台のプリッジまたは構成されているすべてのプリッジに関する情報を表示します。
   - プリッジのリストを表示します。
     ```
     # dladm show-bridge
     ```
   - プリッジのリンク関連のステータスを表示します。
ブリッジの管理（タスクマップ）

# dladm show-bridge -l bridge-name
- ブリッジの統計情報を表示します。

# dladm show-bridge -s bridge-name
- ブリッジのカーネル転送エントリを表示します。

# dladm show-bridge -f bridge-name
- ブリッジに関するTRILL情報を表示します。

# dladm show-bridge -t bridge-name

例 11-1 ブリッジの情報を表示する

次に、さまざまなオプションを付けたdladm show-bridgeコマンドの使用例を示します。

次のコマンドは、システムに構成されているすべてのブリッジに関する情報を表示します。

```
# dladm show-bridge
```

<table>
<thead>
<tr>
<th>BRIDGE</th>
<th>PROTECT</th>
<th>ADDRESS</th>
<th>PRIORITY</th>
<th>DESROOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>goldengate</td>
<td>stp</td>
<td>32768/0:20:bf:f</td>
<td>32768</td>
<td>8192/0:d0:0:76:14:38</td>
</tr>
<tr>
<td>baybridge</td>
<td>stp</td>
<td>32768/0:20:e5:8</td>
<td>32768</td>
<td>8192/0:d0:0:76:14:38</td>
</tr>
</tbody>
</table>

次のdladm show-bridge -l コマンドは、単一のブリッジインスタンス towerに関
するリンク関連のステータス情報を表示します。構成されているパラメータを表
示するには、代わりにdladm show-linkpropコマンドを使用します。

```
# dladm show-bridge -l tower
```

<table>
<thead>
<tr>
<th>LINK</th>
<th>STATE</th>
<th>UPTIME</th>
<th>DESROOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>hme0</td>
<td>forwarding</td>
<td>117</td>
<td>8192/0:d0:0:76:14:38</td>
</tr>
<tr>
<td>qfe1</td>
<td>forwarding</td>
<td>117</td>
<td>8192/0:d0:0:76:14:38</td>
</tr>
</tbody>
</table>

次のdladm show-bridge -s コマンドは、指定されたブリッジterabithiaに関する
統計情報を表示します。

```
# dladm show-bridge -s terabithia
```

<table>
<thead>
<tr>
<th>BRIDGE</th>
<th>DROPS</th>
<th>FORWARDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>terabithia</td>
<td>0</td>
<td>302</td>
</tr>
</tbody>
</table>

次のdladm show-bridge -ls コマンドは、指定されたブリッジlondon上のすべてのリンクに関する統計情報を表示します。

```
# dladm show-bridge -ls london
```

<table>
<thead>
<tr>
<th>LINK</th>
<th>DROPS</th>
<th>RECV</th>
<th>XMIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>hme0</td>
<td>0</td>
<td>360832</td>
<td>31797</td>
</tr>
<tr>
<td>qfe1</td>
<td>0</td>
<td>322311</td>
<td>356852</td>
</tr>
</tbody>
</table>

Oracle Solaris管理:ネットワークインタフェースとネットワーク仮想化 2011年12月
次の dladm show-bridge -f コマンドは、指定されたブリッジ avignon のカーネル転送エントリを表示します。

```bash
# dladm show-bridge -f avignon
DEST  AGE  FLAGS  OUTPUT
8:0:20:bc:a7:dc  10.860 --  hme0
8:0:20:bf:f9:69 --  L  hme0
8:0:20:c0:20:26  17.420 --  hme0
8:0:20:e5:86:11 --  L  qfe1
```

次の dladm show-bridge -t コマンドは、指定されたブリッジ key に関する TRILL情報を表示します。

```bash
# dladm show-bridge -t key
NICK  FLAGS  LINK  NEXTHOP
38628 --  london  56:db:46:be:b9:62
58753 L  --  --
```

▼ ブリッジリンクに関する構成情報を表示する方法

dladm show-link 出力には、BRIDGE フィールドが含まれています。リンクがブリッジのメンバーである場合、このフィールドによってメンバーであるブリッジの名前が識別されます。このフィールドはデフォルトで表示されます。リンクがブリッジの一部でない場合、-p オプションが使用されていれば、このフィールドは空になります。それ以外の場合は、フィールドには -- と表示されます。

dladm show-link 出力には、別々のリンクとしてブリッジ可観測性ノードも表示されます。このノードの場合、既存の OVER フィールドにブリッジのメンバーであるリンクが一覧表示されます。

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティーパーブイ』の「管理権限を取得する方法」を参照してください。

2 ブリッジのメンバーであるリンクに関する構成情報を表示します。

```bash
# dladm show-link [-p]
-p オプションを付けると、解析可能な形式で出力されます。
```

▼ ブリッジを作成する方法

この手順では、STP を使用してブリッジを作成するためのデフォルトの方法を示します。ブリッジ作成のオプションの詳細は、dladm(1M) のマニュアルページで dladm create-bridge の説明を参照してください。
プリッジの管理 (タスクマップ)

注 - 代わりに、TRILL を使用してプリッジを作成するには、-P trill を dladm create-bridge コマンド行に追加するか、dladm modify-bridge コマンドを使用して TRILL を有効にします。

dladm create-bridge コマンドは、プリッジインスタンスを作成し、任意で新しいプリッジに1つ以上のネットワークリックを割り当てます。デフォルトではプリッジインスタンスがシステムに存在しないため、Oracle Solaris ではデフォルトでネットワークリンクを作成します。

リンクをプリッジするには、少なくとも1つのプリッジインスタンスを作成する必要があります。各プリッジインスタンスは個別のものです。プリッジにはプリッジ間の転送接続を含めず、リンクは最大で1つのプリッジのメンバーです。

bridge-name は任意の文字列であり、正当な SMF サービスインスタンス名である必要があります。この名前は、エスケープシーケンスが含まれない FMRI コンポーネントです。つまり、空白、ASCII 制御文字、および次の文字が存在してはいけません。

; / ? : @ & = $ % < > # "

default という名前は、SUNW 文字列で始まるすべての名前と同様、予約されています。末尾に数字が付く名前は、「可観測性デバイス」を作成するために予約されています。可観測性デバイスが使用されるため、正当なプリッジインスタンス名は、正当な dtpl(7P) 名が似たようにさらに制限されます。この名前は、英字または下線文字で開始および終了する必要があります。名前の残りの部分には、英数字と下線文字を含めることができます。

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 プリッジを作成します。
# dladm create-bridge [-l link]... bridge-name
- l link オプションは、プリッジにリンクを追加します。指定されたリンクのいずれかを追加できない場合は、コマンドは失敗し、プリッジが作成されないように注意してください。

次の例は、hme0 と qfe1 リンクを接続して、brooklyn プリッジを作成する方法を示しています。

# dladm create-bridge -l hme0 -l qfe1 brooklyn
▼ ブリッジの保護タイプを変更する方法
この手順では、dladm modify-bridge コマンドを使用して、保護タイプを STP から TRILL または TRILL から STP に変更する方法を示します。

● ブリッジの保護タイプを変更します。
# dladm modify-bridge -P protection-type bridge-name
-P protection-type オプションは、使用する保護タイプを指定します。デフォルトの保護タイプは STP (-P stp) です。代わりに、TRILL 保護タイプを使用するには、-P trill オプションを使用します。
次の例は、brooklyn ブリッジの保護タイプをデフォルトの STP から TRILL に変更する方法を示しています:

# dladm modify-bridge -P trill brooklyn

▼ 既存のブリッジに1つ以上のリンクを追加する方法
この手順では、ブリッジインスタンスに1つ以上のリンクを追加する方法を示します。

1つのリンクは、最大1つのブリッジのメンバーになることができます。したがって、別のブリッジインスタンスにリンクを移動する場合は、まず現在のブリッジからリンクを削除してから、別のブリッジに追加する必要があります。

ブリッジに割り当てられたリンクを VLAN、VNIC、またはトンネルにすることはできません。ブリッジに割り当てることができるのは、集約の一部として受け入れ可能なリンク、または集約自体であるリンクのみです。

ブリッジに割り当てられたリンクはすべて、MTU 値が同じである必要がありま
す。Oracle Solaris では、既存のリンクの MTU 値を変更できることに注意してください。この場合、ブリッジを再起動する前に、割り当てられたリンクを削除または変更して MTU 値が一致するまで、ブリッジインスタンスは保守状態になります。

ブリッジに割り当てられたリンクは、Ethernet タイプ(802.3 および 802.11 メディアを含む)である必要があります。

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 既存のブリッジに新しいリンクを追加します。
# dladm add-bridge -l new-link bridge-name
次の例は、既存のブリッジ rialto に qfe2 リンクを追加する方法を示しています。

```bash
# dladm add-bridge -l qfe2 rialto
```

▼ ブリッジからリンクを削除する方法

この手順では、ブリッジインスタンスから 1 つ以上のリンクを削除する方法を示します。ブリッジを削除する場合は、この手順を使用します。ブリッジを削除する前に、まずブリッジのリンクをすべて削除する必要があります。

1 管理者になります。 詳細は、『Oracle Solaris の管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 ブリッジからリンクを削除します。 

```bash
# dladm remove-bridge [-l link]... bridge-name
```

次の例は、ブリッジ charles から hme0, qfe1, および qfe2 リンクを削除する方法を示しています。

```bash
# dladm remove-bridge -l hme0 -l qfe1 -l qfe2 charles
```

▼ システムからブリッジを削除する方法

この手順では、ブリッジインスタンスを削除する方法を示します。ブリッジを削除する前に、まず dladm remove-bridge コマンドを実行して、接続されているリンクを停止する必要があります。240 ページの「ブリッジからリンクを削除する方法」を参照してください。

1 管理者になります。 詳細は、『Oracle Solaris の管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 システムからブリッジを削除します。

```bash
# dladm delete-bridge bridge-name
```

次の例は、まず coronado ブリッジから hme0, qfe1, および qfe2 リンクを削除してから、システムからブリッジ自体を削除する方法を示しています。

```bash
# dladm remove-bridge -l hme0 -l qfe1 -l qfe2 coronado
# dladm delete-bridge coronado
```
リンク集約の管理

この章では、リンク集約を構成および保守するための手順について説明します。これらの手順には、柔軟なリンク名のサポートなどの新機能を利用する手順が含まれます。

リンク集約の概要

Oracle Solarisでは、リンク集約へのネットワークインタフェースの編成をサポートしています。「リンク集約」は、単一の論理的なユニットとして構成されるシステム上の複数のインタフェースで構成されます。リンク集約は「Trunking」とも呼ばれ、IEEE 802.3ad Link Aggregation Standard (http://www.ieee802.org/3/index.html) で定義されています。

IEEE 802.3ad Link Aggregation Standardには、複数の全二重Ethernetリンクの伝送容量を単一の論理リンクに統合する方法が記載されています。このリンク集約グループは、事実上単一のリンクであるかのように扱われます。

次にリンク集約の機能を示します。

- 帯域幅の増加 - 複数のリンクの伝送容量が1つの論理的なリンクに統合されます。
- 自動フェイルオーバーまたは自動フェイルバック - 障害が発生したリンクのトラフィックが集約内の正常なリンクにフェイルオーバーされます。
- 負荷分散 - 受信と送信の両方のトラフィックが、発信元と着信先のMACアドレスまたはIPアドレスなどのユーザーが選択した負荷分散ポリシーに従って分散されます。
- 冗長性のサポート - 並列集約を使用して2つのシステムを構成できます。
- 管理効率の向上 - すべてのインタフェースが単一のユニットとして管理されます。
リンク集約の概要

- ネットワークアドレスプールのアドレスの節約 – 集約全体に1つのIPアドレスを
  割り当てることができます。

リンク集約の基本

基本的なリンク集約のトポロジーには、一連の物理インタフェースで構成された単一
の集約が含まれます。基本的なリンク集約は、次のような状況で使用します。

- 分散された多くのトラフィックを処理するアプリケーションを実行するシステム
  の場合、集約をそのアプリケーションのトラフィック専用で使用できます。

- IPアドレス空間が制限されていながら大容量の帯域幅が必要なサイトの場合、大
  規模なインタフェースの集約でも1つのIPアドレスのみで済みます。

- 内部インタフェースの存在を隠す必要があるサイトの場合、集約のIPアドレスに
  よって、内部インタフェースを外部アプリケーションから隠します。

図12-1 に、有名なWebサイトをホストするサーバーの集約を示します。このサイト
では、インターネット顧客とサイトのデータベースサーバー間の照会トラ
フィックのために帯域幅を増やす必要があります。セキュリティ上の理由
で、サーバー上の各インタフェースの存在を外部アプリケーションから隠す必要が
あります。解決策として、IPアドレス192.168.50.32で集約aggr1を使用します。こ
の集約は、bge0-bge2の3つのインタフェースで構成されます。これらのインタ
フェースは、顧客の照会に答えるためのトラフィックの送信専用で使用されま
す。すべてのインタフェースからのパケットトラフィック上の送信アドレス
は、aggr1のIPアドレスである192.168.50.32です。

図12-1 基本的なリンク集約トポロジー

図 12-2 は、それぞれの集約が構成された2つのシステムを含むローカルネット
ワークを示しています。2つのシステムはスイッチによって接続されています。ス
イッチ経由で集約を実行する必要がある場合は、そのスイッチが集約デノロジを
サポートしている必要があります。このタイプの構成は、高可用性と冗長性を持つシステムを実現するために特に有効です。

図では、システムAがbge0とbge1という2つのインタフェースで構成される集約を使用しています。これらのインタフェースは、集約に入れたポートを介してスイッチに接続されています。システムBは、e1000g0-e1000g3という4つのインタフェースの集約を使用しています。これらのインタフェースもスイッチの集約に入れたポートに接続されています。

図12-2 スイッチを使用したリンク集約のトポロジー

バックツーバックリンク集約

バックツーバックリンク集約のトポロジーには、次の図に示すように、相互に直接ケーブル接続された2台の別個のシステムが含まれます。これらのシステムでは並列集約が実行されます。
リンク集約の概要

図12-3 基本的なバックツーバック集約のトポロジ

ローカルネットワーク

この図では、システム A 上のデバイス bge0 が、システム B 上の bge0 に直接リンクされ、ほかのデバイスも同様にリンクされています。この方法では、システム A とシステム B は、冗長性と高可用性を提供し、さらに両方のシステム間での高速通信をサポートできます。各システムではさらに、ローカルネットワーク内のトラフィックフロー用の ce0 インタフェースも構成されています。

バックツーバックリンク集約のもっとも一般的なアプリケーションはミラー化されたデータベースサーバーです。両方のサーバーを同時に更新する必要があるため、大きな帯域幅、高速のトラフィックフロー、および信頼性が必要になります。バックツーバックリンク集約のもっとも一般的な使用場所としてデータセンターがあります。

ポリシーと負荷分散

リンク集約を使用する予定の場合は、送信トラフィック用のポリシーを定義することを検討してください。このポリシーや、使用可能な集約のリンク全体にパケットを分散する方法を指定し、負荷分散を確立することができます。次に、使用可能な層指定子と集約ポリシーに対するそれらの意味について説明します。

- **L2** – 各パケットの MAC (L2) ヘッダーをハッシュすることで送信リンクを決定します
- **L3** – 各パケットの IP (L3) ヘッダーをハッシュすることで送信リンクを決定します
- **L4** – 各パケットの TCP, UDP, またはほかの ULP (L4) ヘッダーをハッシュすることで送信リンクを決定します

これらのポリシーを任意に組み合わせて使用することもできます。デフォルトのポリシーは L4 です。詳細は、dladm(1M) のマニュアルページを参照してください。

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月

244
集約モードとスイッチ

集約トポロジにスイッチ経由の接続が含まれている場合は、スイッチが「LACP (Link Aggregation Control Protocol)」をサポートするかどうかに注意する必要があります。スイッチがLACPをサポートしている場合は、スイッチと集約のLACPを構成する必要があります。ただし、次のいずれかのLACPの動作「モード」を定義できます。

- オフモード - 集約のデフォルトのモード。「LACPDU」と呼ばれるLACPパケットは生成されません。
- アクティブモード - ユーザーが指定可能な間隔でシステムによってLACPDUが定期的に生成されます。
- 受動モード - システムは、スイッチからLACPDUを受け取った場合のみLACPDUを生成します。集約とスイッチの両方が受動モードで構成されている場合、それらの間でLACPDUを交換することはできません。

構文については、dladm(1M)のマニュアルページとスイッチの製造元のマニュアルを参照してください。

リンク集約の要件

リンク集約の構成には次のような要件があります。

- dladmコマンドを使用して集約を構成する必要があります。
- すべて作成されているインタフェースを集約のメンバーにすることはできません。
- 集約内のすべてのインタフェースは、同じ速度かつ全二重モードで実行されている必要があります。
- EEPROMパラメータlocal-mac-address?内で、MACアドレスの値を「true」に設定する必要があります。手順の詳細は、インタフェースのMACアドレスが一致であることを確認する方法を参照してください。

一部のデバイスは、リンク状態通知をサポートするためのIEEE 802.3ad Link Aggregation Standardの要件を満たしていません。ポートを集約に接続したり、集約から切り離したりするには、このサポートが存在する必要があります。リンク状態通知をサポートしていないデバイスは、dladm create-aggrコマンドの-fオプションを使用してのみ集約できます。このようなデバイスの場合、リンク状態は常にupとして報告されます。-fオプションの使用については、246ページの「リンク集約を作成する方法」を参照してください。
リンク集約のための柔軟な名前

リンク集約は、柔軟な名前を割り当てることができます。任意のわかりやすい名前をリンク集約に割り当てることができます。柔軟な名前またはカスタマイズされた名前についての詳細は、26ページの「ネットワークデバイスとデータリンク名」を参照してください。以前のOracle Solarisリリースでは、集約に割り当てられた「キー」の値によってリンク集約が識別されます。この方法については、リンク集約の概要を参照してください。この方法は引き続き有効ですが、カスタマイズされた名前を使用してリンク集約を識別するようにしてください。

ほかのすべてのデータリンク構成と同様に、リンク集約はdladmコマンドで管理されます。

リンク集約の管理 (タスクマップ)

次の表は、リンク集約を管理するための手順へのリンクを示しています。

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>集約を作成します。</td>
<td>複数のデータリンクから成る集約を構成します。</td>
<td>246ページの「リンク集約を作成する方法」</td>
</tr>
<tr>
<td>集約を変更します。</td>
<td>集約のポリシーとモードを変更します。</td>
<td>249ページの「集約を変更する方法」</td>
</tr>
<tr>
<td>集約を構成するリンクを変更します。</td>
<td>集約の基になるデータリンクの数を増やすか、または減らします。</td>
<td>249ページの「集約にリンクを追加する方法」または250ページの「集約からリンクを削除する方法」</td>
</tr>
<tr>
<td>集約を削除します。</td>
<td>ネットワーク構成からリンク集約を完全に削除します。</td>
<td>251ページの「集約を削除する方法」</td>
</tr>
</tbody>
</table>

▼ リンク集約を作成する方法

始める前に

注-リンク集約は、同一の速度で稼働する全二重のポイントツーポイントリンク上でのみ機能します。集約内のインタフェースがこの要件を満たしていることを確認してください。

集約トポロジ内でスイッチを使用している場合は、スイッチ上で次の操作を行なったことを確認してください。

Oracle Solaris管理:ネットワークインタフェースとネットワーク仮想化・2011年12月

246
リンク集約の概要

- 集約として使用されるようにポートを構成します。
- スイッチがLACPをサポートしている場合は、LACPをアクティブモードまたは
  受動モードで構成します。

1 マネージャになります。
詳細は、『Oracle Solarisの管理: セキュリティーサービス』の「管理権限を取得する
方法」を参照してください。

2 ネットワークデータリンク情報を表示します。
# dladm show-link

3 集約の作成に使用しているリンクを、どのアプリケーションも開いていないことを
確認します。
たとえば、リンク上にIPインタフェースが作成されている場合は、そのインタ
フェースを削除します。

a. リンクがいずれかのアプリケーションで使用されているかどうかを判定するに
  は、dladm show-link 構文またはipadm show-if 構文のどちらかの出力を検査しま
  す。

<table>
<thead>
<tr>
<th>LINK</th>
<th>CLASS</th>
<th>MTU</th>
<th>STATE</th>
<th>BRIDGE</th>
<th>OVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>qfe3</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

データリンクが使用されている場合は、dladm show-link の出力にある STATE
フィールドでそのリンクがup であることが示されます。したがって、

# dladm show-link

b. IPインタフェースを削除するには、次のコマンドを入力します。

# ipadm delete-ip interface

各情報の意味は次のとおりです。

interfacer そのリンク上で作成されているIPインタフェースを指定します。

4 リンク集約を作成します。

# dladm create-aggr [-f] -l link1 -l link2 [...] aggr
リンク集約の概要

- `f` 集約を強制的に作成します。このオプションは、リンク状態通知をサポートしていないデバイスを集約しようとする場合に使用します。

  `linkn` 集約するデータリンクを指定します。

  `aggr` 集約に割り当てる名前を指定します。

5 集約上にIPインタフェースを作成します。
   
   # ipadm create-ip interface

6 IPインタフェースに有効なIPアドレスを構成します。
   
   # ipadm create-addr interface -T static -a IP-address addrobj
   
   ここで、interfaceには集約の名前を指定し、addrobjはinterface/user-defined-stringの命名規則を使用します。

7 作成した集約のステータスを確認します。
   集約の状態はupである必要があります。
   
   # dladm show-aggr

例12-1 リンク集約の作成

この例は、2つのデータリンクsubvideo0とsubvideo1を含むリンク集約を作成するために使用されるコマンドを示しています。この構成は、システムのループトのあとも永続します。

# dladm show-link
   
   LINK  CLASS  MTU  STATE  BRIDGE  OVER
   subvideo0  phys  1500  up    --  ----
   subvideo1  phys  1500  up46  --  ----

# ipadm delete-ip subvideo0
# ipadm delete-ip subvideo1
# dladm create-aggr -l subvideo0 -l subvideo1 video0
# ipadm create-ip video0
# ipadm create-addr -T static -a 10.8.57.50/24 video/v4
# dladm show-aggr
   
   LINK   POLICY  ADDRPOLICY  LACPACTIVITY  LACPTIMER  FLAGS
   video0  L4  auto  off  short  -----

リンク情報を示すと、このリンク集約がリストに含まれています。

# dladm show-link
   
   LINK  CLASS  MTU  STATE  BRIDGE  OVER
   subvideo0  phys  1500  up    --  ----
   subvideo1  phys  1500  up    --  ----
   video0    aggr  1500  up    --  subvideo0, subvideo1
リンク集約の概要

▼ 集約を変更する方法
この手順では、集約の定義に次の変更を加える方法を示します。

- 集約のポリシーの変更
- 集約のモードの変更

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 集約のポリシーを変更します。

`# dladm modify-aggr -P policy-key aggr`
`policy-key 244 ページの「ポリシーと負荷分散」で説明されているように 1 つ以上のポリシー L2, L3, および L4 を表します。`
`aggr ポリシーを変更する集約を指定します。`

3 集約のLACPモードを変更します。

`# dladm modify-aggr -L LACP-mode -T timer-value aggr`
`-L LACP-mode 集約が実行される LACP モードを示します。値は、active、passive、および off です。スイッチ上で受動モードで LACP が実行されている場合は、集約用にアクティブモードに構成したことを確認してください。`
`-T timer-value LACP タイマー値を示します。値は、short または long です。`

例12-2 リンク集約の変更
この例では、集約 video0 のポリシーを L2 に変更したあと、LACP モードをアクティブにする方法を示しています。

`# dladm modify-aggr -P L2 video0`
`# dladm modify-aggr -L active -T short video0`
`# dladm show-aggr`
`LINK POLICY ADDRPOLICY LACPACTIVITY LACPTIMER FLAGS`
`video0 L2 auto active short -----`

▼ 集約にリンクを追加する方法

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

第12章・リンク集約の管理 249
2 追加するリンクに、そのリンク上で plumb されているIPインタフェースが含まれていないことを確認します。

   # ipadm delete-ip interface

3 集約にリンクを追加します。

   # dladm add-aggr -l link [-l link] [...] aggr
   ここで、link は集約に追加しているデータリンクを表します。

4 さらに多くのデータリンクが追加されたあと、リンク集約機能全体を変更するためのほかのタスクを実行します。
   たとえば、図12-3に示されている構成の場合は、ケーブル接続を追加または変更したり、追加のデータリンクを収容できるようにスイッチを再構成したりすることが必要になる場合があります。スイッチに対して何らかの再構成タスクを実行するには、スイッチのマニュアルを参照してください。

例12-3 集約へのリンクの追加

この例は、集約 video0 にリンクを追加する方法を示しています。

   # dladm show-link
   LINK CLASS MTU STATE BRIDGE OVER
   subvideo0 phys 1500 up -- ----
   subvideo1 phys 1500 up -- ----
   video0 aggr 1500 up -- subvideo0, subvideo1
   net3 phys 1500 unknown -- ----

   # ipadm delete-ip video0
   # dladm add-aggr -l net3 video0
   # dladm show-link
   LINK CLASS MTU STATE BRIDGE OVER
   subvideo0 phys 1500 up -- ----
   subvideo1 phys 1500 up -- ----
   video0 aggr 1500 up -- subvideo0, subvideo1, net3
   net3 phys 1500 up -- ----

▼ 集約からリンクを削除する方法

1 管理者になります。
   詳細は、『Oracle Solarisの管理:セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 集約からリンクを削除します。

   # dladm remove-aggr -l link aggr-link
例 12-4 集約からのリンクの削除

この例では、集約 video0 からリンクを削除する方法を示しています。

dladm show-link
<table>
<thead>
<tr>
<th>LINK</th>
<th>CLASS</th>
<th>MTU</th>
<th>STATE</th>
<th>OVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>subvideo0</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
</tr>
<tr>
<td>subvideo1</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
</tr>
<tr>
<td>video0</td>
<td>aggr</td>
<td>1500</td>
<td>up</td>
<td>--</td>
</tr>
<tr>
<td>net3</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
</tr>
</tbody>
</table>

# dladm remove-aggr -l net3 video0
# dladm show-link
<table>
<thead>
<tr>
<th>LINK</th>
<th>CLASS</th>
<th>MTU</th>
<th>STATE</th>
<th>BRIDGE</th>
<th>OVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>subvideo0</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>----</td>
</tr>
<tr>
<td>subvideo1</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>----</td>
</tr>
<tr>
<td>video0</td>
<td>aggr</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>subvideo0, subvideo1</td>
</tr>
<tr>
<td>net3</td>
<td>phys</td>
<td>1500</td>
<td>unknown</td>
<td>--</td>
<td>----</td>
</tr>
</tbody>
</table>

▼ 集約を削除する方法

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 集約上に構成されている IP インタフェースを削除します。
   # ipadm delete-ip IP-aggr
   ここで、IP-aggr はリンク集約上の IP インタフェースです。

3 リンク集約を削除します。
   # dladm delete-aggr aggr

例 12-5 集約の削除

この例では、集約 video0 を削除します。この削除は永続します。

# ipadm delete-ip video0
# dladm delete-aggr video0
この章では、仮想ローカルエリアネットワーク (VLAN) を構成および保守するための手順について説明します。これらの手順には、柔軟なリンク名のサポートなどの機能を利用する手順が含まれます。

仮想ローカルエリアネットワークの管理

「仮想ローカルエリアネットワーク (VLAN)」は、ローカルエリアネットワークをTCP/IPプロトコルスタックのデータリンク層で分割したものです。スイッチテクノロジを使用するローカルエリアネットワークのVLANを作成できます。ユーザーのグループをVLANに割り当てることで、ローカルネットワーク全体のネットワーク管理とセキュリティを改善できます。さらに、同じシステム上のインタフェースを異なるVLANに割り当てることもできます。

次の作業を行う必要がある場合は、ローカルネットワークをVLANに分割することを検討してください。

- 作業グループの論理的な分割を作成する。
  たとえば、ある建物の1つの階に置かれたすべてのホストが1つのスイッチベースのローカルネットワークに接続されているとします。この階の各作業グループ用に個別のVLANを作成できます。

- 作業グループに異なるセキュリティポリシーを適用する。
  たとえば、財務部門と情報技術部門のセキュリティニーズはまったく異なります。両方の部門のシステムが同じローカルネットワークを共有している場合、各部門用の個別のVLANを作成できます。その後で、適切なセキュリティポリシーをVLANごとに適用できます。

- 作業グループを管理可能なブロードキャストドメインに分割する。
  VLANを使用すると、ブロードキャストドメインのサイズが小さくなり、ネットワークの効率が向上します。
VLAN トポロジの概要

スイッチ LAN テクノロジを使用すると、ローカルネットワーク上のシステムを VLAN に編成できます。ローカルネットワークを VLAN に分割する前に、VLAN テクノロジをサポートするスイッチを入手する必要があります。VLAN トポロジの設計に応じて、スイッチ上のすべてのポートで単一の VLAN を処理するか、複数の VLAN を処理するように構成できます。スイッチのポートを構成する手順はスイッチの製造元によって異なります。

次の図は、サブネットアドレス 192.168.84.0 を使用するローカルエリアネットワークを示しています。この LAN は、RED、YELLOW、および BLUE という 3 つの VLAN に分割されています。

図 13-1 3 つの VLAN を含むローカルエリアネットワーク
LAN 192.168.84.0 の接続は、スイッチ 1 とスイッチ 2 によって処理されます。財務作業グループのシステムは RED VLAN に含まれています。人事作業グループのシステムは YELLOW VLAN 上にあります。情報技術作業グループのシステムは BLUE VLAN に割り当てられています。

**VLAN を使用したネットワークの統合**

ゾーン上の VLAN を使用すると、スイッチなどの 1 つのネットワーク単位内に複数の仮想ネットワークを構成できます。3 枚の物理 NIC を備えたシステムの次の図について考えてみます。

![図13-2 複数のVLANを含むシステム](image)

第13章・VLANの管理
仮想ローカルエリアネットワークの管理

VLAN を使用しない場合は、特定の機能を実行する異なるシステムを構成し、これからのシステムを個別のネットワークに接続することになります。たとえば、Web サーバーをある LAN に、認証サーバーを別の LAN に、さらにアプリケーションサーバーを 3 番目のネットワークに接続します。VLAN とソーンを使用した場合は、8 つのすべてのシステムを解体し、それらを 1 つのシステム内のソーンとして構成することができます。次に、VLAN タグまたは VLAN ID (VID) を使用して、同じ機能を実行する各ソーンのセットに VLAN を割り当てます。この図で提供される情報は次の表に表すことができます。

<table>
<thead>
<tr>
<th>機能</th>
<th>ソーン名</th>
<th>VLAN 名</th>
<th>VID</th>
<th>IP アドレス</th>
<th>NIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web サーバー</td>
<td>webzone1</td>
<td>web1</td>
<td>111</td>
<td>10.1.111.0</td>
<td>e1000g0</td>
</tr>
<tr>
<td>認証サーバー</td>
<td>authzone1</td>
<td>auth1</td>
<td>112</td>
<td>10.1.112.0</td>
<td>e1000g0</td>
</tr>
<tr>
<td>アプリケーションサーバー</td>
<td>apzone1</td>
<td>app1</td>
<td>113</td>
<td>10.1.113.0</td>
<td>e1000g0</td>
</tr>
<tr>
<td>Web サーバー</td>
<td>webzone2</td>
<td>web2</td>
<td>111</td>
<td>10.1.111.0</td>
<td>e1000g1</td>
</tr>
<tr>
<td>認証サーバー</td>
<td>authzone2</td>
<td>auth2</td>
<td>112</td>
<td>10.1.112.0</td>
<td>e1000g1</td>
</tr>
<tr>
<td>アプリケーションサーバー</td>
<td>apzone2</td>
<td>app2</td>
<td>113</td>
<td>10.1.113.0</td>
<td>e1000g1</td>
</tr>
<tr>
<td>Web サーバー</td>
<td>webzone3</td>
<td>web3</td>
<td>111</td>
<td>10.1.111.0</td>
<td>e1000g2</td>
</tr>
<tr>
<td>認証サーバー</td>
<td>authzone3</td>
<td>auth3</td>
<td>112</td>
<td>10.1.112.0</td>
<td>e1000g2</td>
</tr>
</tbody>
</table>

この図に示されている構成を作成するには、例 13–1 を参照してください。

### VLAN のわかりやすい名前

Oracle Solaris では、VLAN インタフェースにわかりやすい名前を割り当てることができます。VLAN 名は、リンク名と VLAN ID 番号 (VID) で構成されます (sales0 など)。VLAN を作成する場合は、カスタマイズされた名前を割り当てるようにしてくださ。カスタマイズされた名前についての詳細は、26 ページの「ネットワークデバイスとデータリンク名」を参照してください。有効なカスタマイズされた名前についての詳細は、30 ページの「有効なリンク名の規則」を参照してください。

### VLAN の管理 (タスクマップ)

次の表は、VLAN を管理するための各種のタスクへのリンクを示しています。
ネットワーク上の VLAN の計画

ネットワーク上の VLAN を計画するには、次の手順に従います。

▼ VLAN 構成を計画する方法

1 ローカルネットワークのトポロジーを調べて、VLAN への分割が適切かどうかを判断します。
   このようなトポロジーの基本的な例については、図 13-1 を参照してください。

2 VID の番号指定スキーマを作成し、各 VLAN に VID を割り当てます。

注 - VLAN の番号指定スキーマは、ネットワーク上にすでに存在している場合があります。その場合は、既存の VLAN 番号指定スキーマに従って VID を作成する必要があるかもしれません。

3 各システム上で、特定の VLAN のメンバーにするインタフェースを決定します。
   a. システム上で構成されているインタフェースを調べます。
      # dladm show-link
   b. システム上の各データリンクに関連付ける VID を判別します。
   c. dladm create-vlan コマンドを使用して VLAN を作成します。
仮想ローカルエリアネットワークの管理

4 インタフェースとネットワークのスイッチの接続を確認します。
各インタフェースと各インタフェースが接続されているスイッチポートの VID を書き留めます。

5 スイッチの各ポートの VID をポートが接続されるインタフェースと同じ VID に構成します。
構成手順については、スイッチの製造元のマニュアルを参照してください。

**VLAN の構成**

次の手順は、VLAN を作成および構成する方法を示しています。Oracle Solaris では、すべての Ethernet デバイスが VLAN をサポートできます。ただし、一部のデバイスではいくつかの制限が存在します。これらの例外については、262 ページの「レガシー・デバイス上の VLAN」を参照してください。

▼ **VLAN を構成する方法**

始める前に VLAN を作成するには、データリンクがシステム上にすでに構成されている必要があります。183 ページの「IP インタフェースを構成する方法」を参照してください。

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 システムで使用されているリンクのタイプを判定します。
# dladm show-link

3 データリンク上に VLAN リンクを作成します。
# dladm create-vlan -l link -v VID vlan-link
link VLAN インタフェースの作成に使用するリンクを指定します。
VID VLAN ID 番号を示します。
vlan-link VLAN の名前を指定します。管理用に選択された名前を指定することもできます。

4 VLAN 構成を確認します。
# dladm show-vlan

5 VLAN 上に IP インタフェースを作成します。
# ipadm create-ip interface
ここで、interface は VLAN 名を使用します。
6 IP インタフェースに IP アドレスを構成します。

```
# ipadm create-addr -T static -a IP-address addrobj
```

ここで、addrobj は interface/user-defined-string の命名規則を使用します。

例 13-1 VLAN の構成

この例では、図 13-2 に示されている VLAN 構成を作成します。この例では、システム内に異なるゾーンがすでに構成されていることを前提にしています。ゾーンの構成についての詳細は、「Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)」のパート II「Oracle Solaris ゾーン」を参照してください。

```
global# dladm show-link
LINK CLASS MTU STATE BRIDGE OVER
e1000g0 phys 1500 up -- --
e1000g1 phys 1500 up -- --
e1000g2 phys 1500 up -- --
global# dladm create-vlan -l e1000g0 -v 111 web1
global# dladm create-vlan -l e1000g0 -v 112 auth1
global# dladm create-vlan -l e1000g0 -v 113 app1
global# dladm create-vlan -l e1000g1 -v 111 web2
global# dladm create-vlan -l e1000g1 -v 112 auth2
global# dladm create-vlan -l e1000g1 -v 113 app2
global# dladm create-vlan -l e1000g2 -v 111 web3
global# dladm create-vlan -l e1000g2 -v 112 auth3
global# dladm show-vlan
LINK VID OVER FLAGS
web1 111 e1000g0 ----
auth1 112 e1000g0 ----
app1 113 e1000g0 ----
web2 111 e1000g1 ----
auth2 112 e1000g1 ----
app2 113 e1000g1 ----
web3 111 e1000g2 ----
auth3 113 e1000g2 ----
```

リンク情報を表示すると、これらの VLAN がリストに含まれています。

```
global# dladm show-link
LINK CLASS MTU STATE BRIDGE OVER
e1000g0 phys 1500 up -- --
e1000g1 phys 1500 up -- --
e1000g2 phys 1500 up -- --
web1 vlan 1500 up -- e1000g0
auth1 vlan 1500 up -- e1000g0
app1 vlan 1500 up -- e1000g0
web2 vlan 1500 up -- e1000g1
auth2 vlan 1500 up -- e1000g1
app2 vlan 1500 up -- e1000g1
web3 vlan 1500 up -- e1000g2
```
仮想ローカルエリアネットワークの管理

auth3  vlan  1500  up  --  e1000g2

これらのVLANをそれぞれ対応するゾーンに割り当てます。たとえば、個々のゾーンに関するネットワーク情報を確認すると、ゾーンごとに次のようなデータが表示されます。

global# zonecfg -z webzone1 info net
net:
  address not specified
  physical: web1

global# zonecfg -z authzone1 info net
net:
  address not specified
  physical: auth1

global# zonecfg -z appzone2 info net
net:
  address not specified
  physical: app2

プロパティ physical の値は、特定のゾーンのために設定されている VLAN を示します。

各非大域ゾーンにログインして、VLAN に IP アドレスを構成します。

webzone1 では:

webzone1# ipadm create-ip web1
webzone1# ipadm create-addr -T static -a 10.1.111.0/24 web1/v4

webzone2 では:

webzone2# ipadm create-ip web2
webzone2# ipadm create-addr -T static -a 10.1.111.0/24 web2/v4

webzone3 では:

webzone3# ipadm create-ip web3
webzone3# ipadm create-addr -T static -a 10.1.111.0/24 web3/v4

authzone1 では:

authzone1# ipadm create-ip auth1
authzone1# ipadm create-addr -T static -a 10.1.112.0/24 auth1/v4

authzone2 では:

authzone2# ipadm create-ip auth2
authzone2# ipadm create-addr -T static -a 10.1.112.0/24 auth2/v4

authzone3 では:

Oracle Solaris管理:ネットワークインタフェースとネットワーク仮想化・2011年12月
仮想ローカルエリアネットワークの管理

authzone3# ipadm create-ip auth3
authzone3# ipadm create-addr -T static -a 10.1.112.0/24 auth3/v4

appzone1 では:

appzone1# ipadm create-ip appl
appzone1# ipadm create-addr -T static -a 10.1.113.0/24 appl/v4

appzone2 では:

appzone2# ipadm create-ip app2
appzone2# ipadm create-addr -T static -a 10.1.113.0/24 app2/v4

▼ リンク集約上にVLANを構成する方法
インタフェース上にVLANを構成する場合と同じ方法で、リンク集約上にVLANを作成することもできます。リンク集約については、第12章「リンク集約の管理」で説明されています。ここでは、VLANとリンク集約の構成について説明します。

始める前に 最初にリンク集約を作成し、次にそのリンク集約に有効なIPアドレスを構成します。リンク集約を作成する場合は、246ページの「リンク集約を作成する方法」を参照してください。

1 システムで構成されている集約を一覧表示します。
   # dladm show-link

2 集約上に作成するVLANごとに、次のコマンドを発行します。
   # dladm create-vlan -l link -v VID vlan-link
   各情報の意味は次のとおりです。
   link  VLANインタフェースの作成に使用するリンクを指定します。この特定のケースでは、リンクはリンク集約を示します。
   VID VLAN ID番号を示します。
   vlan-link VLANの名前を指定します。管理用に選択された名前を指定することもできます。

3 VLAN上にIPインタフェースを作成します。
   # ipadm create-ip interface
   ここで、interfaceはVLAN名を使用します。

4 VLAN上のIPインタフェースに有効なIPアドレスを構成します。
   # ipadm create-addr -T static -a IP-address addrobj
   ここで、addrobjはvlan-int/user-defined-stringの命名規則に従う必要があります。

第13章・VLANの管理 261
例13-2 リンク集約上に複数のVLANを構成する
この例では、リンク集約上に2つのVLANを構成します。VLANにはVID 193と194がそれぞれ割り当てられます。

```
# dladm show-link
LINK     CLASS     MTU  STATE  BRIDGE  OVER
subvideo0 phys 1500 up -- --
subvideo1 phys 1500 up -- --
video0    aggr 1500 up -- subvideo0, subvideo1
```

# dladm create-vlan -l video0 -v 193 salesregion1
# dladm create-vlan -l video0 -v 194 salesregion2

# ipadm create-ip salesregion1
# ipadm create-ip salesregion2

# ipadm create-addr -T static -a 192.168.10.5/24 salesregion1/v4static
# ipadm create-addr -T static -a 192.168.10.25/24 salesregion2/v4static

レガシーデバイス上のVLAN
特定のレガシーデバイスは、最大フレームサイズが1514バイトのパケットのみを処理します。フレームサイズがこの上限を超えるパケットは破棄されます。このような場合は、258ページの「VLANを構成する方法」に示されているのと同じ手順に従います。ただし、VLANを作成するときは、VLANを強制的に作成するために-fオプションを使用します。

実行する一般的な手順は次のとおりです。
1. -fオプションを使用してVLANを作成します。
   ```
   # dladm create-vlan -f -l link -v VID [vlan-link]
   ```
2. 最大転送単位(MTU)に小さなサイズ(1496バイトなど)を設定します。
   ```
   # dladm set-linkprop -p default_mtu=1496 vlan-link
   ```
   MTU値を小さくすることによって、リンク層で転送前にVLANヘッダーを挿入するための領域が確保されます。
3. VLAN内の各ノードのMTUのサイズに同じ小さな値を設定するために同じ手順を実行します。
   リンクプロタビジョン値の変更についての詳細は、157ページの「データリンクの構成(タスク)」を参照してください。
**VLAN** 上でのその他の管理タスクの実行

この節では、新しいdladm サブコマンドのその他の VLAN タスクでの使用法について説明します。また、これらのdladmコマンドはリンク名でも動作します。

▼ VLAN 情報を表示する方法

1 管理者になります。
詳細は、「Oracle Solaris の管理：セキュリティーバス」の「管理権限を取得する方法」を参照してください。

2 VLAN 情報を表示します。

```
# dladm show-vlan [vlan-link]
```

VLAN リンクを指定しない場合は、構成されているすべての VLAN に関する情報が表示されます。

例 13-3 VLAN 情報の表示

次の例は、図13-2で示されている複数の VLAN を含むシステムに基づいており、システム内の使用可能な VLAN を示しています。

```
# dladm show-vlan
LINK   VID  OVER  FLAGS
web1   111  e1000g0 ----
auth1  112  e1000g0 ----
app1   113  e1000g0 ----
web2   111  e1000g1 ----
auth2  112  e1000g1 ----
app2   113  e1000g1 ----
web3   111  e1000g2 ----
auth3  113  e1000g2 ----
```

構成されている VLAN は、dladm show-link コマンドを発行した場合にも表示されます。このコマンド出力で、VLAN は CLASS 列で適切に識別されています。

```
# dladm show-link
LINK  CLASS  MTU  STATE  BRIDGE  OVER
web1  vlan   1500 up   --  e1000g0
auth1 vlan   1500 up   --  e1000g0
app1  vlan   1500 up   --  e1000g0
web2  vlan   1500 up   --  e1000g1
auth2 vlan   1500 up   --  e1000g1
app2  vlan   1500 up   --  e1000g1
web3  vlan   1500 up   --  e1000g2
auth3 vlan   1500 up   --  e1000g2
```
▼ VLAN を削除する方法

1 管理者になります。
詳細は、「Oracle Solaris の管理: セキュリティサービス」の「管理権限を取得する方法」を参照してください。

2 どの VLAN を削除するかを決定します。
   # dladm show-vlan

3 VLAN の IP インタフェースを unplumb します。
   # ipadm delete-ip vlan-interface
   ここで、vlan-interface は VLAN 上に構成されている IP インタフェースです。

注 - 現在使用されている VLAN を削除することはできません。

4 次のいずれかの手順を実行することによって VLAN を削除します。
   ▶ VLAN を一時的に削除するには、次のように -t オプションを使用します。
      # dladm delete-vlan -t vlan
   ▶ 削除を永続的なものにするには、次を実行します。
      a. VLAN を削除します。
         # dladm delete-vlan vlan

例 13-4 VLAN の削除

   # dladm show-vlan
   LINK   VID  OVER   FLAGS
   web1   111  e1000g0 ----
   auth1  112  e1000g0 ----
   app1   113  e1000g0 ----
   web2   111  e1000g1 ----
   auth2  112  e1000g1 ----
   app2   113  e1000g1 ----
   web3   111  e1000g2 ----
   auth3  113  e1000g2 ----

   # ipadm delete-ip web1
   # dladm delete-vlan web1
カスタマイズされた名前を使用しているときのネットワーク構成タスクの結合

この節では、カスタマイズされた名前を使用しているときのリンク、リンク集約、およびVLANの構成に関する、前の章のすべての手順を組み合わせた例について説明します。カスタマイズされた名前を使用するほかのネットワーク構成プラクティスについても、http://www.oracle.com/us/sun/index.htmにある記事を参照してください。

例13-5 リンク、VLAN、およびリンク集約の構成

この例では、4枚のNICを使用するシステムで、8つの個別のサブネットのルーターとして構成する必要があります。この目的を達成するために、サブネットごとに1つずつ、8つのリンクが構成されます。最初に、4枚のすべてのNIC上でリンク集約が作成されます。このタグなしリンクが、デフォルトルートが指すネットワークのデフォルトのタグなしサブネットになります。

次に、ほかのサブネットのために、リンク集約上でVLANインタフェースが構成されます。これらのサブネットは、色分けされたスキームに基づいて名前が付けられます。それに応じて、VLAN名も同様に、それぞれ対応するサブネットに従って名前が付けられます。最終的な構成は、8つのサブネットに対する8つのリンク、つまり1つのタグなしリンクと7つのタグ付きVLANリンクで構成されています。

これらの構成をリブートのあとも永続させるようにするために、以前のOracle Solarisリリースの場合と同じ手順が適用されます。たとえば、IPアドレスを/etc/inet/ndpd.confなどの構成ファイルに追加する必要があります。または、インタフェースの毎回分け規則を規則ファイルに含める必要があります。この例には、これらの最終的な手順は含まれていません。これらの手順については、『Oracle Solarisの管理:IPサービス』の該当する章、特に「TCP/IPの管理」および「DHCP」を参照してください。

```
# dladm show-link
LINK  CLASS  MTU  STATE  BRIDGE  OVER
nge0  phys  1500  up    --    --
nge1  phys  1500  up    --    --
e1000g0 phys 1500  up    --    --
e1000g1 phys 1500  up    --    --

# dladm show-phys
LINK  MEDIA  STATE  SPEED  DUPLEX  DEVICE
nge0  Ethernet up  1000Mb full nge0
nge1  Ethernet up  1000Mb full nge1
e1000g0 Ethernet up  1000Mb full e1000g0
e1000g1 Ethernet up  1000Mb full e1000g1

# ipadm delete-ip nge0
# ipadm delete-ip nge1
# ipadm delete-ip e1000g0
```
例13-5 リンク、VLAN、およびリンク集約の構成（続き）

# ipadm delete-ip e1000g1
# dladm rename-link nge0 net0
# dladm rename-link nge1 net1
# dladm rename-link e1000g0 net2
# dladm rename-link e1000g1 net3

# dladm show-link
<table>
<thead>
<tr>
<th>LINK</th>
<th>CLASS</th>
<th>MTU</th>
<th>STATE</th>
<th>BRIDGE</th>
<th>OVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>net0</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>net1</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>net2</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>net3</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

# dladm show-phys
<table>
<thead>
<tr>
<th>LINK</th>
<th>MEDIA</th>
<th>STATE</th>
<th>SPEED</th>
<th>DUPLEX</th>
<th>DEVICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>net0</td>
<td>Ethernet</td>
<td>up</td>
<td>1000Mb</td>
<td>full</td>
<td>nge0</td>
</tr>
<tr>
<td>net1</td>
<td>Ethernet</td>
<td>up</td>
<td>1000Mb</td>
<td>full</td>
<td>nge1</td>
</tr>
<tr>
<td>net2</td>
<td>Ethernet</td>
<td>up</td>
<td>1000Mb</td>
<td>full</td>
<td>e1000g0</td>
</tr>
<tr>
<td>net3</td>
<td>Ethernet</td>
<td>up</td>
<td>1000Mb</td>
<td>full</td>
<td>e1000g1</td>
</tr>
</tbody>
</table>

# dladm create-aggr -P L2,L3 -l net0 -l net1 -l net2 -l net3 default0

# dladm show-link
<table>
<thead>
<tr>
<th>LINK</th>
<th>CLASS</th>
<th>MTU</th>
<th>STATE</th>
<th>BRIDGE</th>
<th>OVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>net0</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>net1</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>net2</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>net3</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>default0</td>
<td>aggr</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>net0 net1 net2 net3</td>
</tr>
</tbody>
</table>

# dladm create-vlan -v 2 -l default0 orange0
# dladm create-vlan -v 3 -l default0 green0
# dladm create-vlan -v 4 -l default0 blue0
# dladm create-vlan -v 5 -l default0 white0
# dladm create-vlan -v 6 -l default0 yellow0
# dladm create-vlan -v 7 -l default0 red0
# dladm create-vlan -v 8 -l default0 cyan0

# dladm show-link
<table>
<thead>
<tr>
<th>LINK</th>
<th>CLASS</th>
<th>MTU</th>
<th>STATE</th>
<th>BRIDGE</th>
<th>OVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>net0</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>net1</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>net2</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>net3</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>default0</td>
<td>aggr</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>net0 net1 net2 net3</td>
</tr>
<tr>
<td>orange0</td>
<td>vlan</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>default0</td>
</tr>
<tr>
<td>green0</td>
<td>vlan</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>default0</td>
</tr>
<tr>
<td>blue0</td>
<td>vlan</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>default0</td>
</tr>
<tr>
<td>yellow0</td>
<td>vlan</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>default0</td>
</tr>
<tr>
<td>red0</td>
<td>vlan</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>default0</td>
</tr>
<tr>
<td>cyan0</td>
<td>vlan</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>default0</td>
</tr>
</tbody>
</table>

# dladm show-vlan
例13-5 リンク、VLAN、およびリンク集約の構成 （続き）

<table>
<thead>
<tr>
<th>LINK</th>
<th>VID</th>
<th>OVER</th>
<th>FLAGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>orange0</td>
<td>2</td>
<td>default0</td>
<td>-----</td>
</tr>
<tr>
<td>green0</td>
<td>3</td>
<td>default0</td>
<td>-----</td>
</tr>
<tr>
<td>blue0</td>
<td>4</td>
<td>default0</td>
<td>-----</td>
</tr>
<tr>
<td>white0</td>
<td>5</td>
<td>default0</td>
<td>-----</td>
</tr>
<tr>
<td>yellow0</td>
<td>6</td>
<td>default0</td>
<td>-----</td>
</tr>
<tr>
<td>red0</td>
<td>7</td>
<td>default0</td>
<td>-----</td>
</tr>
<tr>
<td>cyan0</td>
<td>8</td>
<td>default0</td>
<td>-----</td>
</tr>
</tbody>
</table>

`# ipadm create-ip orange0`
`# ipadm create-ip green0`
`# ipadm create-ip blue0`
`# ipadm create-ip white0`
`# ipadm create-ip yellow0`
`# ipadm create-ip red0`
`# ipadm create-ip cyan0`

`# ipadm create-addr -T static -a IP-address orange0/v4`
`# ipadm create-addr -T static -a IP-address green0/v4`
`# ipadm create-addr -T static -a IP-address blue0/v4`
`# ipadm create-addr -T static -a IP-address white0/v4`
`# ipadm create-addr -T static -a IP-address yellow0/v4`
`# ipadm create-addr -T static -a IP-address red0/v4`
`# ipadm create-addr -T static -a IP-address cyan0/v4`
第14章

IPMPの紹介

IP ネットワークマルチパス (IPMP) は、ある特定のローカルエリアネットワーク (LAN) に接続された複数のインタフェースを備えたシステムに対し、物理インタフェースの障害検出、透過的なネットワークアクセスフェイルオーバー、およびパケット負荷分散を提供します。

この章では、次の内容について説明します。
- 269 ページの「IPMP の新機能」
- 270 ページの「IPMP の配備」
- 280 ページの「Oracle Solaris の IPMP コンポーネント」
- 281 ページの「IPMP インタフェース構成のタイプ」
- 282 ページの「IPMP アドレス指定」
- 284 ページの「IPMP での障害および修復の検出」
- 288 ページの「IPMP と動的再構成」
- 291 ページの「IPMP の用語と概念」

注 - この章と第15章「IPMP の管理」に含まれる IPMP の説明内の、「インタフェース」という用語へのすべての言及は、特に「IP インタフェース」を意味しています。ネットワークインタフェースカード (NIC) のように、修飾語がこの用語の異なる使用法を明示的に示す場合を除き、この用語は常に、IP 層で構成されたインタフェースを指します。

IPMP の新機能

次の各機能が、現在の IPMP 実装と以前の実装との違いです。
- IPMP グループは IPMP IP インタフェースとして表現されます。このインタフェースは、ネットワークスキャップの IP 層のほかのインタフェースとまったく同様に扱われます。IP 管理タスク、経路指定テーブル、アドレス解決プロトコル (ARP) テーブル、ファイアウォール規則、およびその他の IP 関連手順はすべて、IPMP インタフェースを参照することで IPMP グループを操作します。
IPMPの配備

この節は、IPMPグループの使用に関するさまざまなトピックについて説明します。

IPMPを使用しなければならない理由

さまざまな要因によりインタフェースが使用不可能になる可能性があります。一般に、IPインタフェースは故障する可能性があります。また、インタフェースは、ハードウェア保守のためにオフラインに切り替えられる可能性もあります。そのような場合、IPMPグループを使用しないと、その使用不可能になったインタフェースに関連付けられた他のIPアドレスを使用しても、システムと通信できなくなることになります。さらに、それらのIPアドレスを使用する既存の接続が切断されます。

IPMPを使用すると、1つ以上のIPインタフェースを1つの「IPMPグループ」に構成できます。このグループは、ネットワークトラフィックを送受信するデータアドレス付きのIPインタフェースのように機能します。グループ内のベースとなるインタフェースの1つが故障すると、グループ内の残りのアクティブなベースとなるイ
IPMPの配備

インタフェースの間でデータアドレスが再分配されます。したがって、インタフェースの1つが故障しても、グループはネットワークの接続性を維持します。IPMPでは、グループで最低1つのインタフェースが使用可能であれば、ネットワーク接続を常に使用できます。

また、IPMPは、IPMPグループ内のインタフェースセット全体にアウトバウンドネットワークトラフィックを自動的に分散させることにより、全体的なネットワークパフォーマンスを向上させます。このプロセスは、アウトバウンド「負荷分散」と呼ばれ、システムはさらに、アプリケーションによって発信元IPアドレスが指定されなかったパケットに対して発信元アドレス選択を実行することにより、インバウンド負荷分散も間接的に制御します。ただし、アプリケーションが発信元IPアドレスを明示的に選択した場合は、システムはその発信元アドレスを変更しません。

IPMPを使用する必要がある場合

IPMPグループの構成はシステムの構成によって決まります。次の規則に従います。

- 同じローカルエリアネットワーク(LAN)上の複数のIPインタフェースは、1つのIPMPグループに構成される必要があります。LANは大まかに、「同じリンク層ブロードキャストドメイン」に属するノードを含むVLANや有線/無線の両ローカルネットワークなど、さまざまなローカルネットワーク構成を指します。

注: 同じリンク層(L2)ブロードキャストドメイン上の複数のIPMPグループはサポートされていません。通常、L2ブロードキャストドメインは特定のサブネットに対応します。したがって、サブネットごとにIPMPグループを1つだけ構成する必要があります。

- IPMPグループのベースとなるIPインタフェースが異なるLANにまたがってはいけません。

たとえば、3つのインタフェースを備えたシステムが2つの別のLANに接続されているとします。一方のLANに2つのIPインタフェースがリンクし、他方に単一のIPインタフェースが接続します。この場合、最初の規則により、1つ目のLANに接続する2つのIPインタフェースは1つのIPMPグループとして構成される必要があります。2番目の規則のため、2つ目のLANに接続する単一のIPインタフェースがそのIPMPグループのメンバーになることはできません。その単一のIPインタフェースでは、IPMP構成は必須ではありません。ただし、その単一のインタフェースの可用性を監視するために、そのインタフェースを1つのIPMPグループに構成してもかまいません。単一インターフェースのIPMP構成の詳細については、281ページの「IPMPインタフェース構成のタイプ」を参照してください。
別のケースとして、1つ目のLANへのリンクが3つのIPインタフェースから構成され、もう1つのリンクが2つのインタフェースから構成される場合を考えます。この設定は2つのIPMPグループの構成を必要とします。1つ目のLANにリンクする、3つのインタフェースから成るグループと、2つ目に接続する、2つのインタフェースから成るグループです。

**IPMPとリンク集約の比較**

IPMPとリンク集約は、改善されたネットワークパフォーマンスの実現とネットワーク可用性の維持のための異なるテクノロジです。一般に、高いネットワークパフォーマンスを得るためにリンク集約を配備し、高い可用性を確保するためにIPMPを使用します。

次の表は、リンク集約とIPMPの全般的な比較を示します。

<table>
<thead>
<tr>
<th>IPMP</th>
<th>リンク集約</th>
</tr>
</thead>
<tbody>
<tr>
<td>ネットワークテクノロジのタイプ</td>
<td>レイヤー3（IP層）</td>
</tr>
<tr>
<td>構成ツール</td>
<td>ipadm</td>
</tr>
<tr>
<td>リンクベースの障害検出</td>
<td>サポートされています。</td>
</tr>
<tr>
<td>検査信号ベースの障害検出</td>
<td>ICMP ベースであり、介在するレイヤー2スイッチの複数のレベルにわたって検査用アドレスと同じIPサブネット内で定義された任意のシステムをターゲットとします。</td>
</tr>
<tr>
<td>スタンバイインタフェースの使用</td>
<td>サポートされています</td>
</tr>
<tr>
<td>複数スイッチへのまたがり</td>
<td>サポートされています</td>
</tr>
<tr>
<td>ハードウェアのサポート</td>
<td>必須ではありません</td>
</tr>
<tr>
<td>リンク層の要件</td>
<td>ブロードキャスト可能</td>
</tr>
</tbody>
</table>

Oracle Solaris 管理：ネットワークインタフェースとネットワーク仮想化・2011年12月
<table>
<thead>
<tr>
<th>IPMP</th>
<th>リンク集約</th>
</tr>
</thead>
<tbody>
<tr>
<td>ドライバフレームワークの要件</td>
<td>なし GLDv3フレームワークを使用する必要があります</td>
</tr>
</tbody>
</table>
| 負荷分散のサポート | 存在しており、カーネルによって制御されます。インバウンド負荷分散は、発信元アドレス選択によって間接的に影響されます。
|                    | dladmコマンドを使用した、アウトバウンドトラフィックの負荷分散に対する管理者によるより細粒度の制御。インバウンド負荷分散はサポートされます。

リンク集約では、受信トラフィックは、その集約を構成する複数のリンクにわたって分散されます。したがって、より多くのNICを取り付けて集約にリンクを追加すると、ネットワークのパフォーマンスが向上します。IPMPのトラフィックはIPMPインタフェースのデータアドレスを使用しますが、これは、それらのデータアドレスが、使用可能なアクティブインタフェースにバインドされるからです。たとえば、すべてのデータトラフィックは2つのIPアドレスの間のみを流れるが、それらのトラフィックが同じ接続上を流れるとはかぎらない場合には、より多くのNICを追加してもIPMPのパフォーマンスは改善しません。なぜなら、使用可能なIPアドレスは依然として2つだけだからです。

この2つのデクタロジを互いに補い合うため、両者を一緒に配備して、ネットワークパフォーマンスと可用性の両方の利点を提供できます。たとえば、特定のペンドーによって独自のソリューションが提供されないかぎり、現時点ではリンク集約が複数のスイッチにまたがることはありません。したがって、あるスイッチとホスト間のリンク集約では、そのスイッチがシングルポイント障害となります。スイッチが故障するとリンク集約も同様に失われ、ネットワークのパフォーマンスが低下します。IPMPグループにはこのスイッチの制限はありません。したがって、複数のスイッチを使用するLANのシナリオでは、ホスト上で、それぞれのスイッチに接続するリンク集約をまとめて1つのIPMPグループにすることができます。この構成では、高いネットワークパフォーマンスと高可用性の両方が得られます。あるスイッチが故障すると、その故障したスイッチへのリンク集約のデータアドレスが、グループ内の残りのリンク集約の間で再分配されます。

リンク集約の詳細については、第12章「リンク集約の管理」を参照してください。

IPMP構成での柔軟なリンク名の使用

カスタマイズリンク名のサポートにより、リンクの構成が、そのリンクが関連付けられている物理的なNICに拘束されなくなります。カスタマイズリンク名を使用すると、IPインタフェースを管理する場合の柔軟性が高まります。この柔軟性はIPMPの管理にも及ぶます。あるIPMPグループのベースとなるインタフェースの1つが故障し、交換する必要がある場合や、そのインタフェースを交換する手順が大幅に簡素化されます。交換用のNICが故障したNICと同じタイプである場合、その名前を変更して、故障したNICの構成を継承できます。新しいインタフェースをIPMPグループに追加することもできます。
グループに追加する前に新しい構成を作成する必要はありません。故障した NIC のリンク名を新しい NIC に割り当てるとき、故障したインタフェースと同じ設定で新しい NIC が構成されます。その後、マルチパースデーモンは、アクティブインタフェースとスタンバイインタフェースの IPMP 構成に従ってそのインタフェースを配備します。

したがって、ネットワーク構成を最適化し、IPMP の管理を簡素化するには、インタフェースに汎用名を割り当てることで、インタフェースで柔軟なリンク名を採用する必要があります。次の節 274 ページの「IPMP の動作方法」では、すべての例が、IPMP グループとそのベースとなるインタフェースに柔軟なリンク名を使用しています。カスタマイズリンク名を使用するネットワーク環境で、NIC 交換の背後のプロセスの詳細については、288 ページの「IPMP と動的再構成」を参照してください。ネットワークスタックの概要やカスタマイズリンク名の使用方法については、22 ページの「Oracle Solaris のネットワークスタック」を参照してください。

### IPMP の動作方法

IPMP は、グループが作成されたときのアクティブインタフェースとスタンバイインタフェースの元の数を維持しようとすることによって、ネットワークの可用性を維持します。

グループ内の特定のベースとなる IP インタフェースの可用性を判定するための IPMP 障害検出は、リンクベースまたはプロープベース、あるいはその両方にすることもできます。あるベースとされるインタフェースが故障したと IPMP が判定した場合、そのインタフェースは故障としてフラグが付けられ、使用できなくなります。次に、故障したインタフェースに関連付けられていたデータ IP アドレスが、グループ内で機能している別のインタフェースに再分配されます。さらに、使用可能な場合は、スタンバイインタフェースも配備され、アクティブインタフェースの元の数を維持します。

図 14-1 に示すように、3つのインタフェースを含むアクティブ・スタンバイ構成の IPMP グループ itops0 を考えます。
グループitops0は次のように構成されています。

- このグループには、2つのデータアドレス192.168.10.10と192.168.10.15が割り当てられています。
- ベースとなる2つのインタフェースがアクティブインタフェースとして構成され、柔軟なリンク名net0とnet1が割り当てられています。
- このグループにはスタンバイインタフェースが1つ含まれており、これにも柔軟なリンク名net2が割り当てられています。
- ブローブベースの障害検出が使用されるため、アクティブインタフェースとスタンバイインタフェースは次のような検査用アドレスで構成されています。
  - net0: 192.168.10.30
  - net1: 192.168.10.32
  - net2: 192.168.10.34

注-図のActive、Offline、Reserve、およびFailedの各領域は、ベースとなるインタフェースのステータスを示しているだけであり、物理的な場所を示しているわけではありません。このIPMP実装内では、インタフェースやアドレスの物理的な移動やIPインタフェースの転送は一切発生しません。これらの領域の役割は、ベースとなるインタフェースのステータスが故障、修復のいずれかの結果としてどのように変化するかを示すことだけです。
さまざまなオプションとともに ipmpstat コマンドを使用して、既存の IPMP グループに関する特定の種類の情報を表示できます。その他の例については、322 ページの「IPMP 情報の監視」を参照してください。

図14-1 の IPMP 構成は、次の ipmpstat コマンドを使用して表示できます。

```
# ipmpstat -g
GROUP GROUPNAME STATE FDT INTERFACES
itops0 itops0 ok 10.00s net1 net0 (net2)
```

グループのベースとなるインタフェースに関する情報を表示するには、次のように入力します。

```
# ipmpstat -i
INTERFACE ACTIVE GROUP FLAGS LINK PROBE STATE
net0 yes itops0 ------- up ok ok
net1 yes itops0 --mb--- up ok ok
net2 no itops0 is----- up ok ok
```

IPMP は、アクティブインタフェースの元の数を維持できるようにベースとなるインタフェースを管理することで、ネットワークの可用性を維持します。したがって、net0 が故障すると、グループが引き続き 2 つのアクティブインタフェースを持てるように、net2 が配備されます。net2 のアクティブ化を図14-2 に示します。
ipmpstat ユーティリティは、図14-2の情報を次のように表示します。

```
# ipmpstat -i
INTERFACE ACTIVE GROUP FLAGS LINK PROBE STATE
net0 no itops0 ------- up failed failed
net1 yes itops0 --mb-- up ok ok
net2 yes itops0 -s----- up ok ok
```

net0 が修復されると、アクティブインターフェースとしてのステータスに戻ります。一方、net2 は元のスタンバイステータスに戻されます。

別の故障シナリオを図14-3に示します。このシナリオでは、スタンバイインターフェース net が故障し(1)，その後，アクティブインターフェースの1つである net1 が管理者によってオフラインに切り替えられます(2)。その結果，このIPMPグループには，機能しているインターフェース net0 が1つ残されます。
ipmpstatユーティリティは、図14-3に示された情報を次のように表示します。

```
# ipmpstat -i
INTERFACE ACTIVE  GROUP   FLAGS  LINK  PROBE  STATE
net0   yes    itops0   ------  up     ok     ok
net1   no     itops0   -mb-d-  up     ok     offline
net2   no     itops0   is-----  up     failed failed
```

この特定の故障では、インタフェースが修復された後の回復の動作は異なります。この復元は、修復後の構成と比較したIPMPグループのアクティブインタフェースの元の数に依存します。この回復プロセスを、図14-4に視覚的に示します。
IPMPの配備

図14-4 IPMPの回復プロセス

図14-4 で、net2が修復されると、通常、スタンバイインタフェースとしての元のステータスに戻ります(1)。ところが、このIPMPグループは依然として、アクティブインタフェースの元の数である2個を反映していません。なぜなら、net1が引き続きオフラインのままになっているからです(2)。したがって、IPMPは代わりにnet2をアクティブインタフェースとして配備します(3)。

`ipmpstat` ユーティリティは、修復後のIPMPシナリオを次のように表示します。

```
# ipmpstat -i
INTERFACE  ACTIVE  GROUP  FLAGS  LINK  PROBE  STATE
net0       yes     itops0  -------  up    ok     ok
net1       no      itops0  --mb-d-- up    ok offline
net2       yes     itops0  --s----- up    ok     ok
```

FAILBACK=noモードも構成されたアクティブインタフェースが故障に関与している場合にも、同様の復元シーケンスが発生します。その場合、故障したアクティブインタフェースが修復されても、自動的にアクティブステータスには戻りません。図14-2のnet0がFAILBACK=noモードに構成されているとします。そのモードでは、修復されたnet0は、最初はアクティブインタフェースであったとしても、スタンバイインタフェースとしての予約ステータスに切り替えられます。このIPMPグループのアクティブインタフェースの元の数である2個を維持するように、インタフェースnet2はアクティブのままになります。`ipmpstat` ユーティリティは、次のように回復情報を表示します。

第14章・IPMPの紹介 279
Oracle SolarisのIPMPコンポーネント

Oracle Solaris IPMPには、次のソフトウェアが必要です。

「マルチパスデーモン」in.mpathdは、インタフェースの障害や修復を検出します。ベースとなるインタフェースで検查用アドレスが構成されている場合、このデーモンは、リンクベースの障害検出とプローブベースの障害検出の両方を実行します。このデーモンは、採用された障害検出手法のタイプに応じて、インタフェース上で適切なフラグを設定またはクリアし、そのインタフェースが故障しているかどうかや修復されたかどうかを示します。オプションとして、IPMPグループに属するように構成されていないインタフェースを含むすべてのインタフェースの可用性を監視するようにこのデーモンを構成することもできます。障害検出については、284ページの「IPMPでの障害および修復の検出」を参照してください。

in.mpathdデーモンは、IPMPグループ内のアクティブインタフェースの指定も制御します。このデーモンは、IPMPグループの作成時に最初に構成されたのと同じ数のアクティブインタフェースを維持しようとします。したがって、in.mpathdは、ベースとなるインタフェースを必要に応じて起動したり停止したりして、管理者が構成したポリシーの一貫性を保ちます。in.mpathdデーモンがベースとなるインタフェースの起動を管理する方法の詳細については、274ページの「IPMPの動作方法」を参照してください。このデーモンの詳細については、in.mpathd(IM)のマニュアルページを参照してください。

「IPカーネルモジュール」は、グループ内で使用可能な一連のIPデータアドレスをグループ内で使用可能な一連のベースとなるIPインタフェースに分配することで、アウトバウンド負荷分散を管理します。さらにこのモジュールは、発信元アドレス選択を実行してインバウンド負荷分散を管理します。このIPモジュールのどちらの役割も、ネットワークトラフィックのパフォーマンスを改善します。

IPMP構成ファイル/etc/default/mpathdは、デーモンの動作の構成に使用されます。たとえば、障害検出するためにターゲットをプローブする時間やプローブするインタフェースを設定することで、デーモンがプローブベースの障害検出を実行する方法を指定できます。また、故障したインタフェースが修復されたあとのステータスを指定することもできます。また、このファイルでは、IPMPグループに属するように構成されたインタフェースだけでなく、システム内のすべてのIPインタ

このタイプの構成の詳細については、288ページの「FAILBACK=noモード」を参照してください。
IPMP インタフェース構成のタイプ

IPMP 構成は、通常同じ LAN に接続された同じシステムの複数の物理インタフェースで構成されます。これらのインタフェースは、次のいずれかの構成の IPMP グループに属することができます。

- アクティブ・アクティブ構成 - ベースとなるインタフェースのすべてがアクティブである IPMP グループ。「アクティブインタフェース」とは、IPMP グループによって現時点で使用可能な IP インタフェースのことです。デフォルトでは、あるベースとなるインタフェースを IPMP グループの一部になるように構成すると、そのインタフェースはアクティブになります。アクティブインタフェースやその他の IPMP 用語に関する追加情報については、291 ページの「IPMP の用語と概念」も参照してください。

- アクティブ・スタンバイ構成 - 少なくとも 1 つのインタフェースが予約インタフェースとして管理上構成されている IPMP グループ。この予約インタフェースは「スタンバイインタフェース」と呼ばれます。スタンバイ IP インタフェースはアイドル状態になっていますが、その構成方法に応じて、その可適用性を追跡するためのマルチバスデーモンによって監視されます。インタフェースによってリンク障害通知がサポートされている場合は、リンクベースの障害検出が使用されます。インタフェースで検査用アドレスが構成されている場合は、プローブベースの障害検出も使用されます。いずれかのアクティブインタフェースが故障すると、スタンバイインタフェースが必要に応じて自動的に配備されます。1 つの IPMP グループには、スタンバイインタフェースが必要な数だけ構成できます。

単独インタフェースをそれ自体の IPMP グループ内で構成することもできます。単独インタフェース IPMP グループは、複数のインタフェースを持つ IPMP グループと同じように動作します。ただし、この IPMP 構成は、ネットワークトラフィックの高可用性を提供しません。ベースとなるインタフェースが故障すると、システムはトラフィックを送受信する機能をすべて失います。単一インタフェースの IPMP グループを構成する目的は、障害検出を使用してインタフェースの可用性を監視することです。インタフェースで検査用アドレスを構成することにより、プローブベースの障害検出を使用してそのインタフェースを追跡するようにデーモンを設定できます。単一インタフェースの IPMP グループ構成は通常、Oracle Solaris Cluster ソフト
IPMPアドレス指定

IPMPアドレス指定は、IPv4ネットワークとIPv4およびIPv6のデュアルスタックネットワークで構成できます。IPMPで構成されたインタフェースは、

- 「データアドレス」とは従来のIPv4アドレスやIPv6アドレスのことであり、ブート時にDHCPサーバーによってIPインタフェースに動的に割り当てられるか、あるいはipadmコマンドを使用して手動で割り当てられます。データアドレスはIPMPインタフェースに割り当てられます。標準のIPv4パケットトラフィック（および該当する場合はIPv6パケットトラフィック）が「データトラフィック」とみなされます。データトラフィックのフローはIPMPインタフェースにホストされたデータアドレスを使用し、それらのトラフィックはそのグループのアクティブインタフェースを通過します。

- 「検査用アドレス」とは、ブローブベースの障害および修復検出を実行するためのin.mpathdデーモンによって使用される、IPMP固有のアドレスのことです。検査用アドレスをもはや、DHCPサーバーによって動的に割り当てることも、ipadmコマンドを使用して手動で割り当てることもできます。データアドレスはIPMPインタフェースに割り当てられます。検査用アドレスをIPMPインタフェースにのみ割り当てることもできます。IPMPインタフェースのグループベースとなるインタフェースの場合は、IPv4検査用アドレスまたはIPv6検査用アドレスが利用されます。あるいはその両方を構成できます。あるベースとなるインタフェースが障害した場合、そのインタフェースの検査用アドレスが引き続きin.mpathdデーモンによってブローブベースの障害検出のために使用され、そのインタフェースがその後修復されたかどうかがチェックされます。
注 - プローブベースの障害検出を特に使用する場合のみ、検査用アドレスを構成する必要があります。それ以外の場合は、推移的プローブを有効にすることで、検査用アドレスを使用しなくても障害を検出できます。検査用アドレスを使用する場合としない場合のプローブベースの障害検出の詳細については、284ページの「検査信号ベースの障害検出」を参照してください。

以前のIPMP実装では、特にインタフェースの故障中にアプリケーションによって使用されないように、検査用アドレスはDEPRECATEDとしてマークされる必要がありました。現在の実装では、検査用アドレスはベースとなるインタフェース内に存在しています。したがって、IPMPを認識しないアプリケーションによってこれらのアドレスが間違って使用されることはありませんでした。ただし、これらのアドレスがデータパケットの発信元の候補として考慮されないように、システムは自動的に、NOFAILOVERフラグの付いたすべてのアドレスをDEPRECATEDとしてもマークします。

IPv4検査用アドレス

一般に、サブネット上の任意のIPv4アドレスを検査用アドレスとして使用できます。IPv4検査用IPアドレスは、ルートが指定できなくても構いません。IPv4アドレスは、多くのサイトでは限定リソースなので、ルート指定できないRFC1918プライベートアドレスを検査用IPアドレスとして指定したい場合もあります。in.mpathdデーモンは、ICMP検査信号を検査用IPアドレスと同じサブネットのホストとしか交換しません。RFC1918形式の検査用アドレスを使用していない場合は、ネットワーク上のほかのシステム（ルーターが望ましい）を適切なRFC1918サブネットのアドレスで必ず構成してください。この構成により、in.mpathdデーモンは、ターゲットシステムと正常に検査信号を交換できます。RFC1918プライベートアドレスの詳細については、RFC1918, Address Allocation for Private Internets (http://www.ietf.org/rfc/rfc1918.txt?number=1918)を参照してください。

IPv6検査用IPアドレス

有効なIPv6検査用IPアドレスは、物理インタフェースのリンクローカルアドレスだけです。IPMP検査用IPアドレスとして機能する別のIPv6アドレスは必要ありません。IPv6リンクローカルアドレスは、インタフェースのメディアアクセスコントロール(MAC)アドレスに基づいています。リンクローカルアドレスは、インタフェースが起動時にIPv6を使用できるようになったり、インタフェースがipadmによって手動で構成されたりした場合に、自動的に構成されます。

リンクローカルアドレスの詳細については、『System Administration Guide: IP Services』の「Link-Local Unicast Address」を参照してください。
IPMPでの障害および修復の検出

IPMPグループですべてのグループのインタフェースにIPv4とIPv6の両方が使用される場合には、個別のIPv4検査用アドレスは必要ない場合があります。in.mpathdデーモンは、IPv6リンクルーカルアドレスを検査用IPアドレスとして使用します。

IPMでの障害および修復の検出

IPMPは、トラフィックを受信するネットワークの継続的可用性を保証するために、IPMPグループのベースとなるIPインタフェースに対して障害検出を実行します。故障したインタフェースは、修復されるまで使用不可能なままになります。残りのアクティブインタフェースが機能し続ける一方で、既存のスタンバイインタフェースが必要に応じて配備されます。

IPMPの障害検出の種類

in.mpathdデーモンは、次の種類の障害検出を処理します。

- プローブベースの障害検出(2種類)
  - 検査用アドレスが構成されない(推移的プローブ)。
  - 検査用アドレスが構成される。
- リンクベースの障害検出(NICドライバがサポートしている場合)

検査信号ベースの障害検出

プローブベースの障害検出では、ICMPプローブを使用してインタフェースが故障しているかどうかをチェックします。この障害検出手法の実装は、検査用アドレスが使用されるかどうかによって決まります。

検査用アドレスを使用しないプローブベースの障害検出

検査用アドレスを使用しないこの手法は、2種類のプローブを使用して実装されています。

- ICMPプローブ

ICMPプローブは、経路指定テーブルに定義されたターゲットをプローブするため、グループ内のアクティブインタフェースによって送信されます。「アクティブ」インタフェースとは、そのインタフェースのリンク層(L2)アドレス宛てのインバウンドIPパケットを受信できるベースとなるインタフェースのことです。ICMPプローブは、データアドレスをそのプローブの発信元アドレスとして使用します。ICMPプローブがそのターゲットに到達し、ターゲットから応答が得られた場合、そのアクティブインタフェースは動作しています。

- 推移的プローブ
推移的プローブは、アクティブインタフェースをプローブするために、グループ内の代替インタフェースによって送信されます。代替インタフェースとは、インバウンド IP パケットを能動的に受信しないベースとなるインタフェースのことで、たとえば、4つのベースとなるインタフェースから成る IPMP グループを考えます。このグループでは、データアドレスは1つ構成されていますが、検査用アドレスは1つも構成されていません。この構成では、アウトバウンドパケットはベースとなるインタフェースをすべて使用できます。一方、インバウンドパケットは、データアドレスがオフラインされたインタフェースによってのみ受信できます。インバウンドパケットを受信できない残り3つのベースとなるインタフェースが、「代替」インタフェースとなります。

代替インタフェースがアクティブインタフェースへのプローブの送信と応答の受信に成功した場合、そのアクティブインタフェースは機能しており、推論により、プローブを送信した代替インタフェースも機能しています。

注 - 検査用アドレスを必要としないこの障害検出手法を使用するには、推移的プローブを有効にする必要があります。

検査用アドレスを使用するプローブベースの障害検出

この障害検出手法では、検査用アドレスを使用する ICMP 検査信号メッセージを送受信します。「プローブトラフィック」またはテストトラフィックとも呼ばれるこれらのメッセージは、インタフェース上から同じローカルネットワーク上の1つ以上のターゲットシステムへと送信されます。デーモンは、プローブベースの障害検出用に構成された全てのインタフェースを経由して全てのターゲットを個別にプローブします。ある特定のインタフェースで、連続する5つの検査信号に対して応答がない場合、in.mpathd はそのインタフェースに障害があるものとみなします。検査信号を発信する頻度は、「障害検出時間」に依存します。障害検出時間のデフォルト値は10秒です。ただし、障害検出時間は IPMP 構成ファイルで調整できます。手順については、319ページの「IPMP デーモンの動作を構成する方法」を参照してください。プローブベースの障害検出を最適化するには、マルチバスデーモンからのプローブを受信する複数のターゲットシステムを設定する必要があります。複数のターゲットシステムを設定することで、報告された障害の性質をより正確に判断できます。たとえば、唯一定義されたターゲットシステムから応答がない場合、そのターゲットシステムの障害を示している可能性もあれば、IPMP グループのインタフェースの1つの障害を示している可能性もあります。これに対し、いくつかのターゲットシステムのうちの1つのシステムだけがプローブに応答しない場合は、IPMP グループ自体ではなく、ターゲットシステムで障害が発生している可能性があります。

in.mpathd デーモンは、プローブするターゲットシステムを動的に決定します。まず、デーモンは経路指定テーブル内で、IPMP グループのインタフェースに関連付けられた検査用アドレスと同じサブネット上有るターゲットシステムを検索し、
IPMPでの障害および修復の検出

す。そのようなターゲットが見つかった場合、デーモンはそれらをプロープのターゲットとして使用します。同じサブネット上でターゲットシステムが見つからない場合、in.mpathdは、リンク上の近くのホストをプロープするマルチキャストパケットを送信します。ターゲットシステムとして使用するホストの決定にあたっては、すべてのホストを意味するマルチキャストアドレス(IPv4では224.0.0.1、IPv6ではff02::1)にマルチキャストパケットが送信されます。エコーパケットに応答する最初の5つのホストが、プロープのターゲットとして選択されます。in.mpathdがマルチキャストプロープに続いてICMPエコーパケットに応答したルーターまたはホストを検出できなかった場合、in.mpathdは検査信号ベースの障害を検出できません。この場合、ipmpstat -iユーティリティはプロープの状態をunknownとして報告します。

ホストルートを使用して、in.mpathdが使用するターゲットシステムのリストを明示的に構成できます。手順については、316ページの「プロープベースの障害検出のための構成」を参照してください。

グループ障害

「グループ障害」は、IPMPグループ内のすべてのインタフェースが同時に故障したと思われる場合に発生します。この場合、ベースとなるインタフェースは一切使用できません。また、すべてのターゲットシステムが同時に故障したときに、プロープベースの障害検出が有効になっていた場合、in.mpathdデーモンはその現在のターゲットシステムをすべてフラッシュし、新しいターゲットシステムに対してプロープします。

検査用アドレスを持たないIPMPグループでは、アクティブインタフェースをプロープできる単一のインタフェースがプローブとして指定されます。この指定されたインタフェースには、FAILEDフラグとPROBERフラグが両方とも設定されます。このインタフェースにデータアドレスがバインドされるため、このインタフェースは引き続き、ターゲットをプロープして回復を検出できます。

リンクベースの障害検出

リンクベースの障害検出は、インタフェースがその種の障害検出をサポートしている場合には、常に有効です。

他社製のインタフェースがリンクベースの障害検出をサポートしているかどうかを判定するには、ipmpstat -iコマンドを使用します。ある特定のインタフェースの出力のLINK列にunknownステータスが含まれる場合、そのインタフェースはリンクベースの障害検出をサポートしません。デバイスに関するより具体的な情報については、製造元のドキュメントを参照してください。

リンクベースの障害検出をサポートするこれらのネットワークドライバは、インタフェースのリンク状態を監視し、リンク状態が変わったときにネットワークサブシステムに通知します。変更を通知されると、ネットワークサブシステムは、インタ

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
フェースのRUNNINGフラグを適宜設定または解除します。インタフェースのRUNNINGフラグが解除されたことを検出すると、in.mpathdデーモンは即座にインタフェースに障害があるものとみなします。

障害検出と匿名グループ機能
IPMPは匿名グループでの障害検出をサポートしています。デフォルトでは、IPMPはIPMPグループに属するインタフェースのステータスのみを監視します。ただし、どのIPMPグループにも属さないインタフェースのステータスも追跡するようにIPMPデーモンを構成することもできます。したがって、これらのインタフェースは「匿名グループ」の一部とみなされます。コマンドipmstat -gを発行した場合、匿名グループは二重ダッシュ(...)として表示されます。匿名グループ内のインタフェースでは、データアドレスが検査用アドレスとしても機能します。これらのインタフェースは名前付きのIPMPグループに属していないため、これらのアドレスはアプリケーションから可視となります。IPMPグループの一部でないインタフェースの追跡を有効にする方法については、319ページの「IPMPデーモンの動作を構成する方法」を参照してください。

物理インタフェースの回復検出
「修復検出時間が」は障害検出時間を2倍です。障害検出のデフォルト時間は10秒です。したがって、修復検出のデフォルト時間は20秒です。故障したインタフェースがRUNNINGフラグで再びマークされ、障害検出手法が再検査済みとして検出すると、in.mpathdはそのインタフェースのFAILEDフラグをクリアします。修復されたインタフェースは、管理者が最初に設定したアクティブインタフェースの数に応じて再配備されます。

あるベースとなるインタフェースが故障したときに、ブロードベースの障害検出が使用されていた場合、in.mpathdデーモンは、検査用アドレスが構成されていない場合は指定されたブロードベースのデーモンが検査用アドレスを使用して、ブロードベースを破壊します。インタフェース修復時の復元は、故障したインタフェースの元の構成に応じて進みます。

故障したインタフェースが最初アクティブインタフェースだった- 修復されたインタフェースは元のアクティブステータスに戻ります。システム管理者によって定義された数のインタフェースがそのグループでアクティブになっているときは、障害中に代用品として機能していたスタンバイインタフェースは元のスタンバイインタフェースに切り替えられます。

注- この手順の例外は、修復されたアクティブインタフェースがFAILBACK=noモードでも構成されていた場合です。詳細については、288ページの「FAILBACK=noモード」を参照してください。
IPMPと動的再構成

- 故障したインタフェースが最初スタンバイインタフェースだった - IPMPグループがアクティブインタフェースの元の数を反映している場合、修復されたインタフェースは元のスタンバイステータスに戻ります。それ以外の場合、スタンバイインタフェースはアクティブインタフェースに切り替えられます。

インタフェースの故障や修復時のIPMPの動作方法のグラフィカル表現を確認するには、274ページの「IPMPの動作方法」を参照してください。

FAILBACK=no モード

デフォルトでは、故障したあと修復されたアクティブインタフェースは自動的にグループ内で元のアクティブインタフェースに戻ります。この動作は、デーモンの構成ファイル内のFAILBACKパラメータの設定によって制御されます。ただし、管理者によっては、データアドレスが修復されたインタフェースに再マッピングされるときに発生する短い中断でも許容できない可能性もあります。そうした管理者は、起動されたスタンバイインタフェースが引き続きアクティブインタフェースとして機能できるようにすることを好む可能性があります。IPMPでは、管理者がデフォルト動作を上書きして、インタフェースが修復時に自動的にアクティブにならないようにすることができます。これらのインタフェースはFAILBACK=noモードで構成する必要があります。関連する手順については、319ページの「IPMPデーモンの動作を構成する方法」を参照してください。

FAILBACK=noモードのアクティブインタフェースが故障したあと修復された場合、IPMPデーモンはIPMPの構成を次のように復元します。

- IPMPグループがアクティブインタフェースの元の構成を反映している場合、デーモンはこのインタフェースのINACTIVEステータスを維持します。
- 修復時点でのIPMPの構成が、グループのアクティブインタフェースの元の構成を反映していない場合、FAILBACK=noステータスであるにもかかわらず、修復されたインタフェースがアクティブインタフェースとして再配布されます。

注 - FAILBACK=NOモードはIPMPグループ全体に対して設定されます。これは、インタフェース単位でチューニング可能なパラメータではありません。

IPMPと動的再構成

動的再構成(DR)機能によって、システムの実行中にインタフェースなどのシステムハードウェアを再構成できます。DRは、この機能をサポートするシステムでのみ使用できます。

一般的には、cfgadmコマンドを使用して、DR操作を実行します。ただし、ほかの方法で動的再構成を行うプラットフォームもあります。必ず使用するプラットフォームのドキュメントを参照し、DRの実行に関する詳細を確認してください。
い。Oracle Solaris を使用するシステムの場合は、DR に関する特定のドキュメントは、表 14-1 に記載したリソースで見つけることができます。DR に関する最新情報も http://www.oracle.com/technetwork/jp/indexes/documentation/index.html から入手可能となっており、「動的再構成 (dynamic reconfiguration)」というトピックで検索することで情報を取得できます。

表 14-1 動的再構成の文書リソース

<table>
<thead>
<tr>
<th>説明</th>
<th>参照先</th>
</tr>
</thead>
<tbody>
<tr>
<td>cfgadm コマンドの詳細情報</td>
<td>cfgadm(1M) のマニュアルページ</td>
</tr>
<tr>
<td>Oracle Solaris Cluster 環境での DR に関する具体的な情報</td>
<td>『Oracle Solaris Cluster システム管理』</td>
</tr>
<tr>
<td>Oracle 製 Sun サーバーの DR に関する具体的な情報</td>
<td>特定のサーバーに付属するドキュメントを参照</td>
</tr>
<tr>
<td>DR と cfgadm コマンドに関する紹介情報</td>
<td>『Oracle Solaris の管理: デバイスとファイルシステム』の第 6 章「デバイスの動的構成 (手順)」</td>
</tr>
<tr>
<td>DR をサポートするシステムでの IPMP グループの管理タスク</td>
<td>320 ページの「動的再構成を使用した IPMP 構成の回復」</td>
</tr>
</tbody>
</table>

次の各節は、DR が IPMP とどのように連携するかについて説明します。

NIC の DR をサポートするシステム上では、IPMP を使用して接続を保持したり、既存の接続の切断を防止できます。IPMP は、RCM (Reconfiguration Coordination Manager) フレームワークと統合されています。したがって、NIC の接続、切り離し、または再接続を安全に行うことができ、RCM がシステムコンポーネントの動的再構成を管理します。

新しい NIC の接続

DR がサポートされている場合、新しいインタフェースを接続して plumb したあと、それを既存の IPMP グループに追加できます。また、適当であれば、新たに追加したインタフェースを独自の IPMP グループで構成することも可能です。IPMP グループの構成手順については、304 ページの「IPMP グループの構成」を参照してください。これらのインタフェースが構成されると、これらのインタフェースはすぐに IPMP で使用可能となります。ただし、カスタマイズリンク名を使用するメリットの恩恵を受けるには、インタフェースのハードウェアベースのリンク名の代わりとなる汎用リンク名を割り当てる必要があります。次に、その割り当てたばかりの汎用名を使用して、対応する構成ファイルを作成します。カスタマイズリンク名を使用して単一のインタフェースを構成する手順については、183 ページの「IP インタフェースを構成する方法」を参照してください。汎用リンク名をインタフェースに割り当てたら、インタフェースを IPMP で使用する場合など、インタフェース上で何らかの追加構成を実行する場合は常に汎用名を参照するようにしてください。
NICの切断

NICを含むシステムコンポーネントを切断するすべての要求は、まず接続性を保持できるかどうかチェックされます。たとえば、デフォルトでは、IPMPグループ内のNICを切断することはできません。IPMPグループ内の機能中のインタフェースだけを含むNICも切断できません。ただし、システムコンポーネントを削除しなければならない場合は、cfdadm(1M)のマニュアルページに説明されているcfdadmの-iオプションを使用して、この動作を無効にできます。

チェックが成功すると、デーモンはインタフェースにOFFLINEフラグを設定します。インタフェース上の検査用アドレスがすべて構成解除されます。次に、NICはシステムをunplumbします。これらの手順のいずれかが失敗した場合、または同じシステムコンポーネントのその他のハードウェアのDRで障害が発生した場合は、前の構成が元の状態にリストアされます。このイベントに関するステータスメッセージが表示されます。それ以外の場合、切断要求は正常に完了しています。システムからコンポーネントを削除できます。既存の接続は切断されません。

NICの交換

IPMPグループのベースとなるインタフェースの1つが故障した場合、典型的な解決策は、新しいNICを接続して故障したインタフェースを交換することです。RCMは、実行中のシステムから切断されたNICに関連する構成情報を記録します。故障したNICを「同一」のNICと交換した場合、RCMは、以前にipadmコマンドを使用して定義された永続的な構成に従って、インタフェースを自動的に構成します。

たとえば、故障したbge0インタフェースを別のbge0インタフェースと交換するとします。ipadmコマンドを使用して定義された、故障したbge0の構成設定は、永続的な設定です。交換用のbge NICを接続すると、RCMはそのbge0インタフェースをplumbしたあと、これらの永続的な設定に従ってそのインタフェースを構成します。したがって、インタフェースは検査用アドレスで適切に構成され、IPMPグループに追加されます。

故障したNICを別のNICと交換することは、両者が同じタイプ(Ethernetなど)である場合には可能です。この場合、新しいインタフェースが接続されると、RCMはそれを使用します。インタフェースを最初に構成した際にカスタマイズリンク名を使用しなかった場合、IPMPグループに新しいインタフェースを追加する前にそのNICを構成する必要があります。

一方、カスタマイズリンク名を使用していた場合、追加の構成手順は不要です。故障したインタフェースのリンク名を新しいインタフェースに割り当て直すと、その新しいインタフェースは、取り外されたインタフェースの永続的な設定で指定されていた構成を取得します。次に、RCMがそれらの設定に従ってインタフェースを構成します。インタフェースの故障時にDRを使用してIPMP構成を回復する手順については、320ページの「動的再構成を使用したIPMP構成の回復」を参照してください。
IPMP の用語と概念

この節は、このドキュメントのIPMPの章を通して使用される用語と概念を紹介します。

アクティブインタフェース

システムがデータトラフィックを送受信するときに使用できるベースとなるインタフェースを指します。インタフェースは、次の条件が満たされる場合にアクティブになります。

- インタフェース内で少なくとも1つのIPアドレスがUPになっている。UPアドレスを参照してください。
- FAILED、INACTIVE、またはOFFLINEフラグがインタフェースに設定されていない。
- 重複したハードウェアアドレスを持つことを示すフラグがインタフェースに設定されていない。

「使用不可能なインタフェース」、「INACTIVEインタフェース」と比較してください。

データアドレス

データの発信元アドレスまたは宛先アドレスとして使用できるIPアドレスを指します。データアドレスはIPMPグループの一部であり、グループ内の任意のインタフェース上でトラフィックの送受信に使用できます。さらに、グループ内の1つのインタフェースが機能している場合は、IPMPグループのデータアドレスのセットを継続的に使用できます。以前のIPMP実装では、データアドレスは、IPMPグループのベースとなるインタフェース上でホストされていました。現在の実装では、データアドレスはIPMPインタフェース上でホストされています。

DEPRECATEDアドレス

データの発信元アドレスとして使用できないIPアドレスを指します。通常、IPMP検査用アドレスにはNOFAILOVERフラグが付いていますが、システムによって自動的にDEPRECATEDとしてもマークされます。ただし、任意のアドレスにDEPRECATEDのマークを
動的再構成
付けて、そのアドレスが発信元アドレスとして使用されることを防止できます。
進行中の操作にほとんど、またはまったく影響を与えることなく、システムを実行しながらシステムを再構成できるようにする機能を指します。OracleのSunプラットフォームの一部は、DRをサポートしていません。プラットフォームの一部は、特定のタイプのハードウェアのDRだけをサポートする場合があります。NICのDRをサポートするプラットフォーム上では、IPMPを使用して、DRの実行中もシステムへの中断のないネットワークアクセスを提供できます。
IPMPがどのようにDRをサポートするかについては、288ページの「IPMPと動的再構成」を参照してください。

明示的なIPMPインタフェース作成
現在のIPMP実装にのみ適用されます。この用語は、ipadm create-ipmpコマンドを使用してIPMPインタフェースを作成する方法を指します。明示的なIPMPインタフェース作成は、IPMPグループを作成するための推奨の方法です。この方法では、管理者がIPMPインタフェース名やIPMPグループ名を設定できます。
「明示的なIPMPインタフェース作成」と比較してください。

FAILBACK=noモード
ベースとなるインタフェースの、インタフェース修復時の再分配を避けることによりインタフェースへの受信アドレスの再バインドを最小限に抑える設定を指します。具体的には、インタフェースの修復が検出されると、そのインタフェースのFAILEDフラグがクリアされます。ただし、その修復されたインタフェースのモードがFAILBACK=noの場合にはINACTIVEフラグも設定され、そのインタフェースの使用が禁止されます。ただし、機能している2つ目のインタフェースも存在していることが前提となります。IPMPグループ内の2つ目のインタフェースが故障した場合は、INACTIVEインタフェースも使用の
IPMPの用語と概念

対象となります。現在のIPMP実装では
フェイルバックの概念は適用されたくありませんが、管理上の互換性のためにこのモード
の名前が残されています。

**FAILED インタフェース**
in.mpathdデーモンが故障していると判定したインタフェースを示します。この判定
は、リンクベースの障害検出、ブロープ
ベースの障害検出のいずれかによって実現されます。すべての故障したインタフェースに
はFAILEDフラグが設定されます。

**障害検出**
物理インタフェースや、インタフェースから
インターネット層デバイスまでのパスが動作していないことを検出す処理を指しま
す。リンクベースの障害検出とブロープ
ベースの障害検出という、2つの形態の障害
検出が実装されています。

**暗黙的なIPMPインタフェース作成**
ifconfigコマンドを使用してベースとなるイ
ンタフェースを存在しないIPMPグループ内
に配置することでIPMPインタフェースを作
成する方法を指します。暗黙的なIPMPイン
タフェース作成は、以前のOracle Solarisリ
リースのIPMP実装との下位互換性のために
サポートされています。したがって、この方
法は、IPMPインタフェース名やIPMPグ
ループ名を設定する機能は提供しませ
ん。ipadmコマンドでは暗黙的なIPMPイン
タフェース作成はサポートされません。

「明示的なIPMPインタフェース作成」と比
較してください。

**INACTIVE インタフェース**
機能しているが、管理ポリシーに従って使用されていないインタフェースを指します。す
べてのINACTIVEインタフェースにはINACTIVE
フラグが設定されます。

「アクティブインタフェース」、「使用不可
能なインタフェース」と比較してください。

**IPMP匿名グループのサポート**
ネットワークインタフェースがIPMPグ
ループに属するかどうかにかかわらず、シス
テム内のすべてのネットワークインタ
フェースのステータスをIPMPデーモンが追

第14章・IPMPの紹介 293
IPMPグループ

ネットワークの可用性や利用率を改善するためにシステムによって交換可能として扱われる一連のネットワークインタフェースを指します。各IPMPグループには一連のデータアドレスが含まれていますが、システムはそれらのデータアドレスを、グループ内の任意の一連のアクティブインタフェースに関連付けることができます。この一連のデータアドレスの使用は、ネットワーク可用性を維持し、ネットワーク利用率を改善します。管理者は、IPMPグループ内に配置するインタフェースを選択できます。ただし、同じグループ内のすべてのインタフェースは、同じリンクに接続されていることや、同じプロトコルセット (IPv4 と IPv6 など) で構成されていることなど、一連の一般的な特性を共有する必要があります。

IPMPグループインタフェース

「IPMPインタフェース」を参照してください。

IPMPグループ名

IPMPグループの名前を指します。この名前は、ipadm set-ifpropサブコマンドを使用して割り当てることができます。同じIPMPグループ名を持つすべてのベースとなるインタフェースは、同じIPMPグループの一部として定義されます。現在の実装では、IPMPグループ名よりもIPMPインタフェース名に重点が置かれています。管理者には、ipadm create-ipmpサブコマンドを使用してIPMPグループを作成することにより、IPMPインタフェースとIPMPグループの両方で同じ名前を使用することをお勧めします。

IPMPインタフェース

現在のIPMP実装にのみ適用されます。この用語は、ある特定のIPMPグループ、インタフェースのベースとなるインタフェースのいずれかまたはすべて、およびデータアドレスのすべてを表すIPインタフェースを指します。
IPMPの用語と概念

す。現在のIPMP実装では、IPMPインタフェースはIPMPグループを管理するためのコアコンポーネントであり、経路指定テーブル、ARPテーブル、ファイアウォール規則などで使用されます。

**IPMPインタフェース名**

IPMPインタフェースの名前を示します。このマニュアルでは、ipmpNという命名規則を使用しています。システムも、暗黙的なIPMPインタフェースを作成して同じ命名規則を使用します。ただし、管理者は、明示的なIPMPインタフェースを作成して任意の名前を選択できます。

**IPMPシングルトン**

Oracle Solaris Clusterソフトウェアによって使用され、データアドレスが検査用アドレスとしても機能できるようにするIPMP構成を指します。この構成は、たとえば、あるIPMPグループに属するインタフェースが1つだけである場合に適用されます。

**リンクベースの障害検出**

ネットワークカードのリンクステータスを監視することでインタフェースのステータスを判定するパッシブ形式の障害検出を指します。リンクベースの障害検出は、リンクが稼働中かどうかだけをテストします。このタイプの障害検出は、すべてのネットワークカードドライバでサポートされるわけではありません。リンクベースの障害検出は、明示的な構成を必要とせず、リンクの障害の瞬時の検出を提供します。

「プローブベースの障害検出」と比較してください。

**負荷分散**

インバウンドまたはアウトバウンドのトラフィックを一連のインタフェースに分散する処理を指します。負荷分散はロードバランスシングと異なり、負荷が均等に分散されることを保証しません。負荷分散を使用すると、より高いスループットを達成できます。ただし、負荷分散が行われるのは、データが複数の接続を経由して複数の標識に送信される場合だけです。
インバウンド負荷分散は、インバウンドトラフィックをIPMPグループ内の一連のインタフェースにわたって分散させる処理を示します。インバウンド負荷分散は、IPMPで直接制御することはできません。この処理は、発信元アドレス選択アルゴリズムによって間接的に操作されます。

アウトバウンド負荷分散は、アウトバウンドトラフィックをIPMPグループ内の一連のインタフェースにわたって分散させる処理を指します。アウトバウンド負荷分散は、IPモジュールによって宛先単位で実行され、IPMPグループ内のインタフェースのステータスとメンバーに基づいて必要に応じて調整されます。

NOFAILOVERアドレス

以前のIPMP実装にのみ適用されます。ベースとなるインタフェースに関連付けられ、したがってそのベースとなるインタフェースが故障すると使用不可能なままになるアドレスを指します。すべてのNOFAILOVERアドレスにはNOFAILOVERフラグが設定されます。IPMP検査用アドレスはNOFAILOVERとして指定する必要があるのに対し、IPMPデータアドレスは決してNOFAILOVERとして指定してはいけません。このIPMP実装にはフェイルオーバーの概念は存在しません。ただし、NOFAILOVERという用語は、管理上の互換のために残されています。

OFFLINEインタフェース

システムから使用できないように管理上無効化され、通常はシステムから取り外すための準備段階にあるインタフェースを示します。そのようなインタフェースにはOFFLINEフラグが設定されます。if_mpadmコマンドを使用して、インタフェースをオフラインステータスに切り替えることができます。

物理インタフェース

「ベースとなるインタフェース」を参照してください。

ブローブ

pingコマンドで使用されるパケットに似たICMPパケットを指します。このブローブは、ある特定のインタフェースの送受信バスをテストするために使用されます。
ベースの障害検出が有効になっている場合は、in.mpathdデーモンによってプロープパケットが送信されます。プロープパケットは、その発信元アドレスとしてIPMP検査用アドレスを使用します。

検査信号ベースの障害検出

プロープターゲットとプロープを交換してインタフェースのステータスを判定する、アクティブ形式の障害検出を示します。有効にされると、プロープベースの障害検出は各インタフェースの送受信パス全体をテストします。ただし、このタイプの検出では、管理者が明示的に各インタフェースを検査用アドレスで構成する必要があります。

「リンクベースの障害検出」と比較してください。

プロープターゲット

IPMPグループ内のインタフェースと同じリンク上にあるシステムを指します。ターゲットは、プロープベースの障害検出を使用して特定のインタフェースのステータスをチェックできるよう、in.mpathdデーモンによって選択されます。ICMPプロープの送信を行うループ上の任意のホストが、プロープターゲットとなります。通常はルーターがプロープターゲットになります。通常は、障害検出のロジックをプロープターゲット自体の障害から保護するために、いくつかのプロープターゲットが使用されます。

発信元アドレス選択

IPMPグループ内のあるデータアドレスを特定のパケットの発信元アドレスとして選択する処理を指します。使用する発信元アドレスをアプリケーションが特に選択していない場合は常に、システムによって発信元アドレス選択が実行されます。各データアドレスが関連付けられるハードウェアアドレスは1つだけなので、発信元アドレス選択は間接的にインバウンド負荷分散を制御します。

STANDBYインタフェース

グループ内の別のインタフェースが故障した場合にのみ使用されるように管理上構成されます。
IPMPの用語と概念

ターゲットシステム

たインタフェースを示します。すべてのSTANDBYインタフェースにはSTANDBYフラグが設定されます。

検査用アドレス

「ブローブターゲット」を参照してください。

検査信号用の発信元アドレスまたは宛先アドレスとして使用する必要があり、データトラフィック用の発信元アドレスまたは宛先アドレスとして使用してはいけないIPアドレスを指します。検査用アドレスはベースとなるインタフェースに関連付けられます。ベースとなるインタフェースがUP検査用アドレスで構成されている場合、in.mpathdデーモンはブローブベースの障害検出を使用してこのアドレスを監視します。すべての検査用アドレスはNOFAILOVERとして指定する必要があります。また、これらのアドレスはシステムによってDEPRECATEDとして自動的にマークされますが、これは、これらのアドレスがデータパケットの発信元アドレスの候補として考慮されないようにするためです。

ベースとなるインタフェース

IPMPグループの一部であり、実際のネットワークデバイスに直接関連付けられたIPインタフェースを指定します。たとえば、ce0とcelがIPMPグループipmp0内に配置される場合、ipmp0のベースとなるインタフェースはce0とcelから構成されます。以前の実装では、IPMPグループは、ベースとなるインタフェースだけから構成されています。一方、現在の実装では、これらのインタフェースは、グループ(したがってグループ名)を表すIPMPインタフェース(ipmp0など)のベースとなっています。

オフライン取り消し処理

以前オフラインにしたインタフェースをシステムから管理上使用できるようにする行為を指します。if_mpadmコマンドを使用して、オフライン取り消し処理を実行できます。

使用不可能なインタフェース

現在の構成でデータトラフィックの送受信に一切使用できないベースとなるインタフェースを指します。使用不可能なインタフェースは、現在使用されていないが、グ
ループ内のアクティブインタフェースが使用不可能になった場合に使用できるINACTIVEインタフェースとは異なります。次のいずれかの条件が存在する場合にインタフェースが使用不可能になります。

- インタフェースにUPアドレスがない。
- FAILEDまたはOFFLINEフラグがインタフェースに設定されている。
- インタフェースに、グループ内の別のインタフェースと同じハードウェアアドレスを持つことを示すフラグが付いている。

UPアドレス

UPフラグを設定することでシステムから管理上使用可能になったアドレスを指します。UPでないアドレスは、システムに属していないものとして扱われるため、発信元アドレス選択時に考慮されることは決してありません。
第15章

IPMPの管理

この章では、IPネットワークマルチパス(IPMP)でインタフェースグループを管理するためのタスクを紹介します。この章では、主に次の内容について説明します。

- 301ページの「IPMP管理のタスクマップ」
- 304ページの「IPMPグループの構成」
- 312ページの「IPMPグループの維持」
- 316ページの「プローカーベースの障害検出のための構成」
- 320ページの「動的再構成を使用したIPMP構成の回復」
- 322ページの「IPMP情報の監視」

IPMP管理のタスクマップ

Oracle Solarisでは、ipmpstatコマンドが、IPMPグループに関する情報の取得に使用する優秀なツールです。この章では、ipmpstatコマンドが、以前のOracle SolarisリリースでIPMP情報を提供するために使用されていたifconfigコマンドの特定の機能を置き換えています。

ipmpstatコマンドのさまざまなオプションについては、322ページの「IPMP情報の監視」を参照してください。

次の各節は、この章内のタスクへのリンクを提供します。

IPMPグループの作成および構成(タスクマップ)

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPMPグループを計画します。</td>
<td>IPMPグループを構成する以前の補助的な情報と必要なタスクをすべて示す</td>
<td>304ページの「IPMPグループの計画を立てる方法」</td>
</tr>
</tbody>
</table>
### IPMP管理のタスクマップ

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCPを使用してIPMPグループを構成します。</td>
<td>DHCPを使用してIPMPグループを構成する代替手法を提供します。</td>
<td>306ページの「DHCPを使用してIPMPグループを構成する方法」</td>
</tr>
<tr>
<td>アクティブ-アクティブIPMPグループを構成します。</td>
<td>ベースとなるインタフェースのすべてがネットワークトラフィックをホストするように配備されるIPMPグループを構成します。</td>
<td>309ページの「アクティブ-アクティブIPMPグループを手動で構成する方法」</td>
</tr>
<tr>
<td>アクティブ-スタンバイIPMPグループを構成します。</td>
<td>ベースとなるインタフェースの1つが予約として非アクティブ状態に維持されるIPMPグループを構成します。</td>
<td>310ページの「アクティブ-スタンバイIPMPグループを手動で構成する方法」</td>
</tr>
</tbody>
</table>

### IPMPグループの維持(タスクマップ)

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPMPグループへのインタフェースの追加</td>
<td>既存のIPMPグループのメンバーとして新しいインタフェースを追加する</td>
<td>312ページの「IPMPグループにインタフェースを追加する方法」</td>
</tr>
<tr>
<td>IPMPグループからのインタフェースの削除</td>
<td>IPMPグループからインタフェースを削除します。</td>
<td>313ページの「IPMPグループからインタフェースを削除する方法」</td>
</tr>
<tr>
<td>IPMPグループにIPアドレスを追加した後、グループからIPアドレスを削除した後</td>
<td>IPMPグループのアドレスを追加または削除します。</td>
<td>314ページの「IPアドレスを追加または削除する方法」</td>
</tr>
<tr>
<td>インタフェースのIPMPメンバーシップを変更します。</td>
<td>IPMPグループ間でインタフェースを移動する</td>
<td>315ページの「インタフェースを1つのIPMPグループから別のグループに移動する方法」</td>
</tr>
<tr>
<td>IPMPグループを削除します。</td>
<td>不要となったIPMPグループを削除します。</td>
<td>316ページの「IPMPグループを削除する方法」</td>
</tr>
<tr>
<td>故障したカードを交換します。</td>
<td>IPMPグループの故障したNICを取り外すか交換します。</td>
<td>320ページの「障害が発生した物理カードを交換する方法」</td>
</tr>
</tbody>
</table>
プローブベースの障害検出の構成 (タスクマップ)

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>ターゲットシステムを手動で指定します。</td>
<td>プローブベースの障害検出のターゲットとなるシステムを特定して追加します。</td>
<td>317ページの「検査信号ベースの障害検出のターゲットシステムを手動で指定する方法」</td>
</tr>
<tr>
<td>プローブベースの障害検出の動作を構成します。</td>
<td>プローブベースの障害検出の動作を決定するパラメータを変更します。</td>
<td>319ページの「IPMP デーモンの動作を構成する方法」</td>
</tr>
</tbody>
</table>

IPMP グループの監視 (タスクマップ)

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>グループ情報を取得します。</td>
<td>IPMP グループに関する情報を表示します。</td>
<td>322ページの「IPMP のグループ情報を取得する方法」</td>
</tr>
<tr>
<td>データアドレスの情報を取得します。</td>
<td>IPMP グループによって使用されるデータアドレスに関する情報を表示します。</td>
<td>323ページの「IPMP のデータアドレス情報を取得する方法」</td>
</tr>
<tr>
<td>IPMP のインタフェース情報を取得します。</td>
<td>IPMP インタフェースまたは IPMP グループのベースとなるインタフェースに関する情報を表示します。</td>
<td>324ページの「グループのベースとなるIP インタフェースに関する情報を取得する方法」</td>
</tr>
<tr>
<td>プローブターゲットの情報を取得します。</td>
<td>プローブベースの障害検出のターゲットに関する情報を表示します。</td>
<td>326ページの「IPMP のプローブターゲット情報を取得する方法」</td>
</tr>
<tr>
<td>プローブ情報を取得します。</td>
<td>システムの進行中のプローブに関するリアルタイム情報を表示します。</td>
<td>328ページの「IPMP のプローブを監視する方法」</td>
</tr>
<tr>
<td>IPMP グループを監視するための情報の表示をカスタマイズします。</td>
<td>表示される IPMP 情報を決定します。</td>
<td>329ページの「スクリプト内でipmpstat コマンドの出力をカスタマイズする方法」</td>
</tr>
</tbody>
</table>
IPMPグループの構成

この節は、IPMPグループの計画および構成に使用される手順を提供します。第14章「IPMPの紹介」の概要は、IPMPグループの実装をインタフェースとして説明しています。したがって、この章では、用語「IPMPグループ」と「IPMPインタフェース」が同義として使用されます。

▼ IPMPグループの計画を立てる方法

次の手順には、必要となる計画タスクとIPMPグループを構成する前に収集する情報が含まれています。これらのタスクは、順番どおり行う必要はありません。

注 - 各サブネットまたはL2ブロードキャストドメインに対して、IPMPグループを1つだけ構成する必要があります。詳細は、271ページの「IPMPを使用する必要がある場合」を参照してください。

1 ユーザーの要求に適した全般的なIPMP構成を決定します。

IPMPの構成は、システム上でホストされるタイプのトラフィックを処理するためにネットワークに何が必要かによって決まります。IPMPはアウトバウンドのネットワークパケットをIPMPグループのインタフェース間で分散するため、ネットワークのスループットを改善します。ただし、ある程度のTCP接続では、インバウンドトラフィックは通常、アウトオブオーダーのパケットを処理するリスクを最小限に抑えるために、1つの物理パスのみをたどります。

したがって、ネットワークが大量のアウトバウンドトラフィックを処理する場合、多数のインタフェースを1つのIPMPグループに構成すると、ネットワークのパフォーマンスを改善できます。代わりに、システムが大量のインバウンドトラフィックをホストする場合、グループ内のインタフェースの数を増やしても、トラフィックの負荷分散によってパフォーマンスが改善されることはかぎりません。ただし、ベースとなるインタフェースの数を増やせば、インタフェースが故障した際のネットワークの可用性を保証しやすくなります。

2 SPARCベースシステムの場合、グループ内の各インタフェースが一意のMACアドレスを持っていることを確認します。

システム内のインタフェースごとに一意のMACアドレスを構成する方法については、181ページの「SPARC:インタフェースのMACアドレスが一意であることを確認する方法」を参照してください。

Oracle Solaris管理:ネットワークインタフェースとネットワーク仮想化・2011年12月
3 STREAMSモジュールの同じセットが転送され、IPMPグループ内のすべてのインタフェースで構成されていることを確認します。
同じグループのすべてのインタフェースは、同じ順番で構成されたSTREAMSモジュールを持っているかではありません。

a. 予想されるIPMPグループのすべてのインタフェースのSTREAMSモジュールの順番を確認します。
   ifconfig interface modlist コマンドを使用して、STREAMSモジュールの一覧を出力できます。たとえば、net0インタフェースのifconfig出力は次のようになります。
   
   # ifconfig net0 modlist
   0 arp
   1 ip
   2 e1000g
   この出力が示しているように、インタフェースは通常、IPモジュールのすぐ下のネットワークドライバとして存在します。これらのインタフェースでは、追加の構成は必要ないはずです。
   ただし、特定のテクノロジー、IPモジュールとネットワークドライバの間にそのテクノロジーをSTREAMSモジュールとして挿入します。STREAMSモジュールの処理状態を把握可能な場合には、グループ内のすべてのインタフェースに同じモジュールを転送している場合でも、フェイルオーバーで予想外の動作が実行される可能性があります。ただし、IPMPグループのすべてのインタフェースに同じ順番で転送している場合は、処理状態を把握できないSTREAMSモジュールを使用できます。

b. インタフェースのモジュールをIPMPグループでの標準的な順番で転送します。
   
   ifconfig interface modinsert module-name@position
   ifconfig net0 modinsert vpnmod@3

4 IPMPグループのすべてのインタフェースで同じIPアドレス指摘書式を使用します。
1つのインタフェースがIPv4向けに構成されている場合は、そのグループのすべてのインタフェースをIPv4向けに構成する必要があります。たとえば、1つのインタフェースにIPv6アドレス指定を追加すると、IPMPグループ内のすべてのインタフェースをIPv6をサポートするように構成する必要があります。

5 実装する障害検出のタイプを決定します。
たとえば、ブロードベースの障害検出を実装する場合は、ベースとなるインタフェースで検査用アドレスを構成する必要があります。関連情報については、284ページの「IPMPの障害検出の種類」を参照してください。
6 IPMPグループ内のすべてのインターフェースが同じローカルネットワークに接続されていることを確認します。
たとえば、同じIPサブネット上のEthernetスイッチを1つのIPMPグループに構成できます。1つのIPMPグループにいくつもインターフェースを構成できます。

注- たとえば、システムに物理インターフェースが1つだけ存在する場合は、単一インタフェースのIPMPグループに構成することもできます。関連情報については、281ページの「IPMP用インタフェース構成のタイプ」を参照してください。

7 IPMPグループに、別のネットワークメディアタイプのインターフェースが含まれていないことを確認します。
グループ化するインターフェースは、/usr/include/net/if_types.hで定義されている同じインターフェースタイプになるべきです。たとえば、1つのIPMPグループにEthernetインターフェースとトークリングインターフェースを組み合わせることはできません。別の例としては、同じIPMPグループに、トークリングインターフェースと非同期送信モード(ATOM)インターフェースを組み合わせることはできません。

8 ATMインターフェースを持つIPMPの場合は、LANエミュレーションモードでATMインターフェースを構成します。
IPMPは、従来型のIPをATMで使用するインターフェースではサポートされていません。

▼ DHCPを使用してIPMPグループを構成する方法

現在のIPMP実装では、動的ホスト構成プロトコル(DHCP)のサポートを使用してIPMPグループを構成できます。

複数インターフェースのIPMPグループは、アクティブ-アクティブインターフェースで構成することも、アクティブ-スタンバイインタフェースで構成することもできます。関連情報については、281ページの「IPMP用インタフェース構成のタイプ」を参照してください。次の手順は、DHCPを使用してアクティブ-スタンバイIPMPグループを構成する手順について説明します。

始める前に IPMPグループに含める予定のIPインターフェースが、システムのネットワークデタリンク上で正しく構成されていることを確認します。IPMPインターフェースは、ベースとなるIPインターフェースがまだ存在していなくても作成できます。ただし、このIPMPインターフェースでのその後の構成は失敗します。

リンクやIPインターフェースの構成手順については、181ページの「IPインターフェースの構成(タスク)」を参照してください。IPv6インテラフェースの構成方法については、「Oracle Solarisの管理: IPサービス」の「IPv6インタフェースの構成」を参照してください。
さらに、SPARCシステムを使用する場合は、インタフェースごとに一意のMACアドレスを構成します。手順については、181ページの「SPARC: インタフェースのMACアドレスが一意であることを確認する方法」を参照してください。

最後に、DHCPを使用する場合は、ベースとなるインタフェースのリースが無限になっていることを確認します。それ以外の場合、グループで障害が発生した場合には、検査用アドレスが期限切れとなり、IPMPデーモンがプローブベースの障害検出を無効化し、リンクベースの障害検出が使用されます。リンクベースの障害検出によってインタフェースが機能していることがわかると、デーモンは誤ってインタフェースが修復されたと報告する可能性があります。DHCPの構成方法の詳細については、『System Administration Guide: IP Services』の第13章「Planning for DHCP Service (Tasks)」を参照してください。

注 - システムのアクティブネットワークプロファイルがアクティブなプロファイルの場合は、IPMPを使用できません。IPMPグループを構成する前に、必要に応じてDefaultFixedプロファイルを有効化し、固定ネットワーク構成プロファイルに切り替えます。手順については、154ページの「プロファイルと構成ツール」を参照してください。

1 管理者になります。
詳細は、『Oracle Solarisの管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 IPMP インタフェースを作成します。
   # ipadm create-ipmp ipmp-interface
各情報の意味は次のとおりです。

   ipmp-interfaceは、IPMP インタフェースの名前を指定します。IPMP インタフェースには、意味のある任意の名前を割り当てることができます。ほかのIPインタフェースと同様に、名前は ipmp0 のように文字列と数字から構成されます。

3 ベースとなるIP インタフェースがまだ存在していない場合はそれを作成します。
   # ipadm create-ip under-interface
   ここで、under-interfaceは、IPMP グループに追加するIP インタフェースを表します。

4 検査用アドレスを含むベースとなるIP インタフェースを、IPMP グループに追加します。
   # ipadm add-ipmp -i under-interface1 [-i under-interface2 ...] ipmp-interface
   IP インタフェースは、システムで使用可能な数だけIPMP グループに作成できます。
5 IPMPインタフェース上のデータアドレスをDHCPに構成および管理させます。

```
# ipadm create-addr -T dhcp addrobj
```

`addrobj`はアドレスオブジェクトを表し、interface/stringという形式を使用します。interfaceは、この手順ではIPMPインタフェースです。文字列は、任意のユーザー定義文字列です。したがって、IPMPインタフェース上にデータアドレスが複数存在する場合、対応するアドレスオブジェクトは、ipmp-interface/string1、ipmp-interface/string2、ipmp-interface/string3のようにになります。

6 ベースとなるインタフェース内の検査用アドレスをDHCPに管理させます。

IPMPグループのベースとなるインタフェースごとに、次のコマンドを発行する必要があります。

```
# ipadm create-addr -T dhcp addrobj
```

`addrobj`はアドレスオブジェクトを表し、interface/stringという形式を使用します。interfaceは、この手順では、ベースとなるインタフェースです。文字列は、任意のユーザー定義文字列です。したがって、IPMPグループのベースとなるインタフェースが複数存在する場合、対応するアドレスオブジェクトは、under-interface1/string、ipmp-interface2/string、ipmp-interface3/stringのようにになります。

例 15-1 DHCPを使用したIPMPグループの構成

この例は、DHCPを使用してアクティブ-スタンバイIPMPグループを構成する方法を示しますが、次のシナリオに基づいています。

- IPMPグループの3つのベースとなるインタフェースがそれぞれ対応するデータリンクnet0、net1、およびnet2上で構成されます。これらがIPMPグループの指定されたメンバーになります。
- IPMPインタフェースitops0は、同じ名前をIPMPグループと共有します。
- net2が指定されたスタンバイインタフェースになります。
- プローブベースの障害検出を使用するために、ベースとなるインタフェースのすべてに検査用アドレスが割り当てられます。

```
# ipadm create-ipmp itops0
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ip net2
# ipadm add-ipmp -i net0 -i net1 -i net2 itops0
# ipadm create-addr -T dhcp itops0/dhcp0
# ipadm create-addr -T dhcp itops0/dhcp1
# ipadm create-addr -T dhcp net0/test
```
アクティブ-アクティブ IPMPグループを手動で構成する方法

次の手順は、アクティブ-アクティブ IPMPグループを手動で構成する手順について説明します。

1 管理者になります。
詳細は、「Oracle Solaris の管理: セキュリティ・サービス」の「管理権限を取得する方法」を参照してください。

2 IPMPインタフェースを作成します。
   # ipadm create-ipmp ipmp-interface
各情報の意味は次のとおりです。
   ipmp-interface は、IPMPインタフェースの名前を指定します。IPMPインタフェースには、意味のある任意の名前を割り当てることができます。ほかのIPインタフェースと同様に、名前はipmp0のように文字列と数字から構成されます。

3 ベースとなるIPインタフェースをグループに追加します。
   # ipadm add-ipmp -i under-interface1 [-i underinterface2 ...] ipmp-interface
ここで、under-interface は、IPMPグループのベースとなるインタフェースを表します。IPインタフェースは、システムで使用可能な数だけ追加できます。
IPMPグループの構成

注 - デュアルスタック環境では、特定のグループにインタフェースのIPv4インスタンスを配置すると、IPv6インスタンスも自動的に同じグループに配置されます。

4 IPMPインタフェースにデータアドレスを追加します。

```
# ipadm create-addr -T static IP-address addrobj
```

*IP-address* はCIDR表記することができます。

*addrobj* は、命名規則 *ipmp-interface/*any-string を使用する必要があります。したがって、IPMPインタフェースの名前が *ipmp0* の場合、*addrobj* は *ipmp0/dataaddr* とすることができます。

5 ベースとなるインタフェースに検査用アドレスを追加します。

```
# ipadm create-addr -T static IP-address addrobj
```

*IP-address* はCIDR表記することができます。

*addrobj* は、命名規則 *under-interface/*any-string を使用する必要があります。したがって、ベースとなるインタフェースの名前が *net0* の場合、*addrobj* は *net0/testaddr* とすることができます。

注 - 検査用IPアドレスを構成する必要があるのは、特定のインタフェースで検査信号ベースの障害検出を使用する場合だけです。

IPMPグループのすべての検査用IPアドレスには、同じネットワークアドレスを使用してください。すべての検査用IPアドレスは1つのIPサブネットに属していればなりません。

▼ アクティブ - スタンバイ IPMPグループを手動で構成する方法

スタンバイインタフェースの詳細については、281ページの「IPMPインタフェース構成のタイプ」を参照してください。次の手順は、1つのインタフェースが予約して確保されるIPMPグループを構成します。このインタフェースは、グループ内のアクティブインタフェースが故障した場合にのみ配備されます。

1 管理者になります。

詳細は、「Oracle Solarisの管理: セキュリティサービス」の「管理権限を取得する方法」を参照してください。

2 IPMPインタフェースを作成します。

```
# ipadm create-ipmp ipmp-interface
```
IPMPグループの構成

各情報の意味は次のとおりです。

*ipmp-interface* は、IPMP インタフェースの名前を指します。IPMP インタフェースには、意味のある任意の名前を割り当てることができます。ほかの IP インタフェースと同様に、名前は ipmp0 のように文字列と数字から構成されます。

### 3 ベースとなる IP インタフェースをグループに追加します。

```
# ipadm add-ipmp -i under-interface1 [-i underinterface2 ...] ipmp-interface
```

ここで、*under-interface* は、IPMP グループのベースとなるインタフェースを表します。IP インタフェースは、システムで使用可能な数だけ追加できます。

注 - デュアルスタック環境では、特定のグループにインタフェースの IPv4 インスタンスを配置すると、IPv6 インスタンスも自動的に同じグループに配置されます。

### 4 IPMP インタフェースにデータアドレスを追加します。

```
# ipadm create-addr -T static IP-address addrobj
```

*IP-address* は CIDR 表記することができます。

*addrobj* は、命名規則 *ipmp-interface/*any-string を使用する必要があります。したがって、IPMP インタフェースの名前が ipmp0 の場合、*addrobj* は ipmp0/dataaddr とすることができます。

### 5 ベースとなるインタフェースに検査用アドレスを追加します。

```
# ipadm create-addr -T static IP-address addrobj
```

*IP-address* は CIDR 表記することができます。

*addrobj* は、命名規則 *under-interface/*any-string を使用する必要があります。したがって、ベースとなるインタフェースの名前が net0 の場合、*addrobj* は net0/testaddr とすることができます。

注 - 検査用 IP アドレスを構成する必要があるのは、特定のインタフェースで検査信号ベースの障害検出を使用する場合だけです。

IPMP グループのすべての検査用 IP アドレスには、同じネットワークアドレスを使用してください。すべての検査用 IP アドレスは 1 つの IP サブネットに属していなければなりません。

### 6 ベースとなるインタフェースの 1 つをスタバイインタフェースとして構成します。

```
# ipadm set-ifprop -p standby=yes under-interface
```
例 15-2 アクティブ - スタンバイ IPMP グループの構成

この例は、アクティブ - スタンバイ IPMP 構成を手動で作成する方法を示します。この例はまず、ベースとなるインタフェースを作成します。

```bash
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ip net2
# ipadm create-ipmp itops0
# ipadm add-ipmp -i net0 -i net1 -i net2 itops0
# ipadm create-addr -T static -a 192.168.10.10/24 itops0/v4add1
# ipadm create-addr -T static -a 192.168.10.15/24 itops0/v4add2
# ipadm create-addr -T static -a 192.168.85.30/24 net0/test
# ipadm create-addr -T static -a 192.168.85.32/24 net1/test
# ipadm create-addr -T static -a 192.168.85.34/24 net2/test
# ipadm set-ifprop -p standby=yes net2
```

```
# ipmpstat -g
GROUP GROUPNAME STATE FDT INTERFACES
itops0 itops0 ok 10.00s net0 net1 (net2)
```

```
# ipmpstat -t
INTERFACE MODE TESTADDR TARGETS
net0 routes 192.168.10.30 192.168.10.1
net1 routes 192.168.10.32 192.168.10.1
net2 routes 192.168.10.34 192.168.10.5
```

### IPMP グループの維持

この節には、既存の IPMP グループとこれらのグループ内のインタフェースを維持するタスクが含まれます。304 ページの「IPMP グループの構成」の説明に従って、すでに IPMP グループを構成していることが前提です。

#### ▼ IPMP グループにインタフェースを追加する方法

始める前に グループに追加するインタフェースが、グループ内に存在するための制約をすべて満たしていることを確認します。IPMP グループの要件の一覧については、304 ページの「IPMP グループの計画を立てること」を参照してください。

1 管理者になります。
詳細は、「Oracle Solaris の管理: セキュリティーサービス」の「管理権限を取得する方法」を参照してください。
IPMPグループの維持

例 15-3 IPMPグループへのインタフェースの追加

インタフェース net4 をIPMPグループitops0に追加するには、次のコマンドを入力します。

```
# ipadm create-ip net4
# ipadm add-ipmp -i under-interface ipmp-interface
```

▼ IPMPグループからインタフェースを削除する方法

例 15-4 グループからのインタフェースの削除

インタフェース net4 をIPMPグループitops0から削除するには、次のコマンドを入力します。

```
# ipadm remove-ipmp net4 itops0
# ipmpstat -g
GROUP   GROUPNAME  STATE  FDT  INTERFACES
itops0  itops0     ok    10.00s  net0 net1 net4
```
IPアドレスを追加または削除する方法

アドレスを追加する場合は `ipadm create-addr` サブコマンドを、インターフェースからアドレスを削除する場合は `ipadm delete-addr` サブコマンドを使用します。現在のIPMP実装では、検査用アドレスはベースとなるIPインタフェースでホストされ、データアドレスはIPMPインタフェースに割り当てられます。次的手順は、検査用アドレス、データアドレスのいずれかとなるIPアドレスを追加または削除する方法について説明します。

1 管理者になります。
詳細は、『Oracle Solarisの管理:セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 データアドレスを追加または削除します。
   - データアドレスをIPMPグループに追加するには、次のコマンドを入力します。
     ```
     # ipadm create-addr -T static -a ip-address addrobj
     addrobjは、命名規則ipmp-interface/user-stringを使用します。
     ```
   - IPMPグループからアドレスを削除するには、次のコマンドを入力します。
     ```
     # ipadm delete-addr addrobj
     addrobjは、命名規則inder-interface/user-stringを使用します。
     ```

3 検査用アドレスを追加または削除します。
   - IPMPグループのベースとなるインタフェースに検査用アドレスを割り当てるには、次のコマンドを入力します。
     ```
     # ipadm create-addr -T static ip-address addrobj
     ```
   - IPMPグループのベースとなるインタフェースから検査用アドレスを削除するには、次のコマンドを入力します。
     ```
     # ipadm delete-addr addrobj
     ```

例15-5 インタフェースからの検査用アドレスの削除
次の例は、例15-2のitopsの構成を使用しています。この手順は、インターフェースnet1から検査用アドレスを削除します。この例では、検査用アドレスの名前がnet1/test1であることを前提としています。

```
# ipmpstat -t
INTERFACE   MODE TESTADDR   TARGETS
net1         routes 192.168.10.30 192.168.10.1

# ipadm show-addr
ADDROBJ   TYPE STATE ADDR
```
インタフェースを1つのIPMPグループから別のグループに移動する方法

インタフェースが既存のIPMPグループに属している場合は、新しいIPMPグループにインタフェースを配置できます。この場合、現在のIPMPグループからインタフェースを削除する必要はありません。新しいグループに追加されたインタフェースは、既存のIPMPグループから自動的に削除されます。

1 管理者になります。
詳細は、『Oracle Solarisの管理：セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 新しいIPMPグループへインターフェースを移動します。

```
# ipadm add-ipmp -i under-interface ipmp-interface
```
ここで、under-interfaceは、移動するベースとなるインターフェースを表し、ipmp-interfaceは、ベースとなるインターフェースの移動先となるIPMPインタフェースまたはIPMPグループを表します。
インタフェースを新しいグループに追加すると、そのインタフェースは現在のグループから自動的に削除されます。

例15-6 別のIPMPグループへのインターフェースの移動
この例は、グループのベースとなるインターフェースがnet0、net11、およびnet2であることを前提としています。net0をIPMPグループcs-link1に移動するには、次のように入力します。

```
# ipadm add-ipmp -i net0 cs-link1
```
このコマンドは、net0インターフェースをIPMPグループitops0から削除したあと、net0をcs-link1に配置します。
△ IPMP グループを削除する方法
この手順は、特定の IPMP グループが不要になったときに使用します。

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 IPMP グループとベースとなる IP インタフェースを特定します。
   # ipmpstat -g

3 IPMP グループに現在属している IP インタフェースをすべて削除します。
   # ipadm remove-ipmp -i under-interface, -i under-interface, ...] ipmp-interface

注：IPMP インタフェースを正常に削除するには、その IPMP グループの一部として IP インタフェースが存在しているのではいかません。

4 IPMP インタフェースを削除します。
   # ipadm delete-ipmp ipmp-interface
IPMP インタフェースを削除すると、このインタフェースに関連付けられた IP アドレスはすべてシステムから削除されます。

例 15-7 IPMP インタフェースの削除
ベースとなる IP インタフェース net0 と net1 を含むインタフェース itops0 を削除するには、次のコマンドを入力します。

   # ipmpstat -g
   GROUP  GROUPNAME  STATE  FDT  INTERFACES
   itops0  itops0  ok  10.00s  net0 net1
   # ipadm remove-ipmp -i net0 -i net1 itops0
   # ipadm delete-ipmp itops0

プロープベースの障害検出のための構成
検査信頼ベースの障害検出では、284 ページの「検査信頼ベースの障害検出」で説明されているようにターゲットシステムを使用します。プロープベースの障害検出のターゲットを特定するために、in.mpathd デーモンはルーターターゲットモード、マルチキャストターゲットモードの 2 つのモードで動作します。ルーターターゲットモードでは、マルチバスデーモンは経路指定テーブルに定義されたターゲットをプロープします。ターゲットが 1 つも定義されていない場合、このデーモンはマルチ
キャストターゲットモードで動作します。この場合、LAN上の近くのホストをブロープするためにマルチキャストパケットが送出されます。

できれば、`in.mpathd` デーモンがブロープするホストターゲットを設定するようにしてください。一部の IPMP グループでは、デフォルトルーターはターゲットとして十分です。ただし、一部の IPMP グループでは、検査信号ベースの障害検出用に特定のターゲットを設定したほうが良いこともあります。ターゲットを設定するには、経路指定テーブル内にホストのルートをブロープターゲットとして設定します。経路指定テーブルに構成されているすべてのホストルートは、デフォルトルーターの前に一覧化されます。IPMP はターゲットを選択するために、明示的に定義されたホストルートを使用します。したがって、デフォルトルーターを使用するのではなく、ホストのルートを設定して特定のブロープターゲットを構成するようにしてください。

ホストのルートを経路指定テーブルに設定するには、`route` コマンドを使用します。このコマンドで `-p` オプションを使用して、永続的なルートを追加できます。たとえば、`route -p add` は、システムのリブート後も経路指定テーブル内に残るルートを追加します。したがって、-p オプションを使用すると、システムが起動するたびにこれらのルートを作成し直す特殊なスクリプトを一切使用することなしに、永続的なルートを追加できます。ブロープベースの障害検出を最適にかつてで使用するには、ブロープを受信するターゲットを必ず複数設定してください。

後述のサンプル手順は、ブロープベースの障害検出のターゲットへの永続的なルートを追加するための正確な構文を示します。`route` コマンドのオプションの詳細については、`route(1M)` のマニュアルページを参照してください。

ネットワーク上のどのホストが適切なターゲットになるのかの評価では、次の基準を検討します。

- 予想されるターゲットが使用可能で、実行されていることを確認します。IP アドレスの一覧を作成します。
- ターゲットインタフェースが、構成中の IPMP グループと同じネットワークにあることを確認します。
- ターゲットシステムのネットマスクとブロードキャストアドレスは、IPMP グループ内のアドレスと同じでなければなりません。
- ターゲットホストは、検査信号ベースの障害検出を使用しているインタフェースからの ICMP 要求に応答できなければなりません。

▼ 検査信号ベースの障害検出のターゲットシステムを手動で指定する方法

1. 検査信号ベースの障害検出を構成しているシステムにユーザーアカウントでログインします。
ブロープベースの障害検出のための構成

2 検査信号ベースの障害検出のターゲットとして使用される特定のホストにルートを追加します。

```
route -p add -host destination-IP gateway-IP -static
```

ここで、destination-IP と gateway-IP は、ターゲットとして使用されるホストのIPv4アドレスです。たとえば、IPMP グループ itops0 のインタフェースと同じサブネット上のターゲットシステム 192.168.10.137 を指定するには、次のように入力します。

```
route -p add -host 192.168.10.137 192.168.10.137 -static
```

この新しいルートは、システムを再起動するたびに自動的に構成されます。ブロープベースの障害検出用のターゲットシステムへの一時的なルートを定義するだけの場合は、-p オプションを使用しないでください。

3 ターゲットシステムとして使用されるネットワーク上の追加ホストにルートを追加します。

▼ 使用する障害検出手法を選択する方法

デフォルトでは、検査用アドレスを使用しないとブロープベースの障害検出を実行できません。NIC ドライバがリンクベースの障害検出をサポートしている場合、この障害検出も自動的に有効になります。

NIC ドライバによってリンクベースの障害検出がサポートされている場合、この手法を無効にすることはできません。ただし、実装するブロープベースの障害検出のタイプは選択可能です。

1 推移的ブロープのみを使用する場合は、次の手順を実行します。

a. 適切な SMF コマンドを使用して IPMP のプロバティー transitive-probing をオンに切り替えます。

```
# svcconf -s svc:/network/ipmp setprop config/transitive-probing=true
# svcadm refresh svc:/network/ipmp:default
```

このプロバティーの設定方法の詳細については、in.mpathd(1M) のマニュアルページを参照してください。

b. IPMP グループ用に構成された既存の検査用アドレスをすべて削除します。

2 検査用アドレスのみを使用して障害をブロープする場合は、次の手順を実行します。

a. 必要に応じて、推移的ブロープをオフにします。

```
# svcconf -s svc:/network/ipmp setprop config/transitive-probing=false
# svcadm refresh svc:/network/ipmp:default
```
b. IPMP グループのベースとなるインタフェースに検査用アドレスを割り当てます。

▼ IPMP デーモンの動作を構成する方法

IPMP デーモンに関連する次のシステム共通パラメータを設定するには、IPMP 構成ファイル `/etc/default/mpathd` を使用します。

- FAILURE_DETECTION_TIME
- TRACK_INTERFACES_ONLY_WITH_GROUPS
- FAILBACK

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティービース』の「管理権限を取得する方法」を参照してください。

2 `/etc/default/mpathd` ファイルを編集します。
3つのパラメータの1つ以上のデフォルト値を変更します。

a. FAILURE_DETECTION_TIME パラメータの新しい値を入力します。

```
FAILURE_DETECTION_TIME=n
```
ここで、`n` は ICMP 検証がインタフェースの障害が発生していないかどうかを検出する時間(秒単位)です。デフォルトは10秒です。

b. FAILBACK パラメータの新しい値を入力します。

```
FAILBACK=[yes | no]
```
- yes - 値 yes が、IPMP のファイバーバック動作のデフォルトです。障害が発生したインタフェースの修復が検出されると、ネットワークアクセスはこの修復されたインタフェースに復帰します。詳細は、287 ページの「物理インタフェースの回復検出」を参照してください。
- no - 値 no は、データトラフィックが修復されたインタフェースに戻らないことを示します。障害が発生したインタフェースの修復が検出されると、そのインタフェースに INACTIVE フラグが設定されます。このフラグは、現時点でそのインタフェースをデータトラフィックに使用すべきでないことを示します。ただし、そのインタフェースを検査信号トラフィックに使用することはできます。

たとえば、IPMP グループ `ipmp0` が2つのインタフェース `net0` と `net1` から構成されています。`/etc/default/mpathd` ファイルで `FAILBACK=no` パラメータが設定されています。`net0` が故障すると、`FAILED` としてフラグが付けられ使用不可能になります。修復後、このインタフェースに `INACTIVE` としてフラグが付けられますが、`FAILBACK=no` が設定されているため、使用不可能なままとなります。

第15章・IPMPの管理 319
動的再構成を使用した IPMP 構成の回復

net1 が故障し、net0 のみが INACTIVE 状態である場合には、net0 の INACTIVE フラグがクリアされ、このインタフェースが使用可能になります。IPMP グループに同じ INACTIVE 状態のインタフェースがほかにも含まれている場合、net1 の故障時にそれらの INACTIVE インタフェースのいずれか 1 つ（必ずしも net0 とはかぎらない）がクリアされ、使用可能となります。

c. TRACK_INTERFACES_ONLY_WITH_GROUPS パラメータの新しい値を入力します。

TRACK_INTERFACES_ONLY_WITH_GROUPS=[yes | no]

注 - このパラメータと匿名グループ機能については、287 ページの「障害検出と匿名グループ機能」を参照してください。

- yes - 値 yes が、IPMP の動作のデフォルトです。このパラメータを指定した場合、IPMP は、IPMP グループに構成されていないネットワークインタフェースを無視します。
- no - 値 no は、IPMP グループ内に構成されているネットワークインタフェースかどうかにかかわらず、すべてのネットワークインタフェースの障害と修復の検出を設定します。ただし、IPMP グループ内に構成されていないインタフェースで障害や修復が検出されても、IPMP では、そのインタフェースのネットワーク機能を維持するためアクションは一切起動されません。したがって、値 no を指定することは、障害の報告には役立ちますが、ネットワークの可用性を直接向上させることはありません。

3 in.mpathd デーモンを再起動します。

# pkill -HUP in.mpathd

動的再構成を使用した IPMP 構成の回復

この機能は、動的再構成 (DR) をサポートするシステムを管理する手順を記載します。

▼ 障害が発生した物理カードを交換する方法

この手順は、DR をサポートするシステムの物理カードを交換する方法について説明します。この手順は次の条件を前提としています。

- システムのアクティブ NCP は DefaultFixed です。システムのアクティブな NCP が DefaultFixed でない場合の DR の使用方法については、42 ページの「その他の Oracle Solaris ネットワーク技術を使用した NWAM の動作」の「動的再構成とネットワーク構成プロファイル」の節を参照してください。
- システムの IP インタフェースは net0 と net1 です。
- どちらのインタフェースも IPMP グループ itops0 に属しています。
ベースとなるインタフェース net0 には検査用アドレスが含まれています。
ベースとなるインタフェース net0 が故障したため、net0 のカード bge を取り外す必要があります。
bge カードを e1000g カードに交換します。

始める前に
DR の実行手順はシステムのタイプによって変わります。したがって、次が完了していることを確認してください。

- システムが DR をサポートしていることを確認します。
- システムでの DR 手順を説明した適切なマニュアルを参照します。Oracle 製 Sun ハードウェアの場合、DR をサポートするシステムはすべてサーバーです。Sun システムの最新の DR ドキュメントを見つけるには、http://www.oracle.com/technetwork/jp/indexes/documentation/index.html で「動的再構成 (dynamic reconfiguration)」を検索します。

注 - 次の手順の各ステップは、特に IPMP やリンク名の使用に関係する DR の側面のみに言及しています。この手順には、DR を実行するための完全な手順は含まれていません。たとえば、IP 層以外の一部の層では、構成が自動化されていない場合、ATM やその他のサービスなどのために手動構成手順が必要になります。システムの対応する DR ドキュメントに従ってください。

NIC の詳細な交換手順については、173 ページの「動的再構成を使用してネットワークインタフェースカードを交換する方法」を参照してください。

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 適切な DR 手順を実行し、故障した NIC をシステムから取り外します。
たとえば、bge カードを取り外します。

3 交換用の NIC をシステムに接続します。
たとえば、bge カードが占有していたと同じ場所に、e1000g カードを取り付けます。e1000g のデータリンクは、名前 net0 を引き継ぎ、そのデータリンクの構成を継承します。

4 新しい NIC リソースが使用可能になるようにして、DR プロセスを完了します。
たとえば、cfgadm コマンドを使用してこの手順を実行します。詳細は、cfgadm(1M)のマニュアルページを参照してください。

この手順のあと、新しいインタフェースは検査用アドレスで構成され、IPMP グループのベースとなるインタフェースとして追加され、アクティブインタ

第 15 章・IPMP の管理
IPMP 情報の監視

次の各手順は、システム上の IPMP グループのさまざまな側面の監視を可能にする ipmpstat コマンドを使用します。IPMP グループ全体のステータスやそのベースとなるIPインタフェースのステータスを監視できます。グループのデータアドレスを検査し、アドレスの構成を確認することもできます。ipmpstat コマンドを使用すると、障害検出に関する情報を取得されます。ipmpstat コマンドとそのオプションの詳細については、ipmpstat(1M)のマニュアルページを参照してください。

デフォルトでは、ホスト名が存在する場合、数値 IP アドレスではなくホスト名が出力に表示されます。出力に数値 IP アドレスを表示するには、-n オプションを、IPMP グループの特定の情報を表示するためのほかのオプションとともに使用します。

注：次の各手順では、特に明記していないかぎり、ipmpstat コマンドの使用時にシステム管理者の特権は必要ありません。

▼ IPMP のグループ情報を取得する方法

この手順は、システム上のさまざまな IPMP グループのステータス（そのベースとなるインタフェースのステータスも含む）を一覧表示する場合に使用します。ある特定のグループでプローブベースの障害検出が有効になっていると、コマンドはそのグループの障害検出時間も含めます。

IPMP のグループ情報を表示します。

$ ipmpstat -g

<table>
<thead>
<tr>
<th>GROUP</th>
<th>GROUPNAME</th>
<th>STATE</th>
<th>FDT</th>
<th>INTERFACES</th>
</tr>
</thead>
<tbody>
<tr>
<td>itops0</td>
<td>itops0</td>
<td>ok</td>
<td>10.00s</td>
<td>net0 net1</td>
</tr>
<tr>
<td>acctg1</td>
<td>acctg1</td>
<td>failed</td>
<td>--</td>
<td>[net3 net4]</td>
</tr>
<tr>
<td>field2</td>
<td>field2</td>
<td>degraded</td>
<td>20.00s</td>
<td>net2 net5 (net7) [net6]</td>
</tr>
</tbody>
</table>

GROUP
IPMP インタフェースの名前を指定します。匿名グループの場合、このフィールドは空になります。匿名グループの詳細については、in.mpathd(1M)のマニュアルページを参照してください。

GROUPNAME
IPMP グループの名前を指定します。匿名グループの場合、このフィールドは空になります。

STATE
グループの現在のステータスを示します。ステータスは次のいずれかになります。
・okは、IPMPグループのベースとなるインタフェースがすべて使用可能であることを示します。
・degradedは、グループ内のベースとなるインタフェースの一部が使用不可能であることを示します。
・failedは、グループのインタフェースがすべて使用不可能であることを示します。

FDT
障害検出が有効になっている場合に、その障害検出時間を指定します。障害検出が無効になっている場合、このフィールドは空になります。

INTERFACES
グループに属するベースとなるインタフェースを指定します。このフィールドでは、まずアクティブなインタフェースが表示され、次にアクティブでないインタフェースが表示され、最後に使用不可能なインタフェースが表示されます。インタフェースのステータスはその表示形式によって示されます:
- interface(丸括弧で角括弧なし)は、アクティブなインタフェースを示します。アクティブなインタフェースとは、システムでデータトラフィックの送受信に使用されているインタフェースのことです。
- (interface)(丸括弧付き)は、機能しているがアクティブではないインタフェースを示します。このインタフェースは、管理ポリシーの定義に従って未使用になっています。
- [interface](角括弧付き)は、インタフェースが故障しているかオフラインになっているために使用不可能であることを示します。

▼ IPMPのデータアドレス情報を取得する方法

この手順は、データアドレスと各アドレスの所属先グループを表示する場合に使用します。表示される情報には、アドレスがipadm [up-addr/down-addr] コマンドによって切り替えられたかどうかに基づいてどのアドレスが使用可能であるかも含まれます。また、あるアドレスがどのインバウンドまたはアウトバウンドインタフェース上で使用できるかも判定できます。

- IPMPのアドレス情報を表示します。

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>STATE</th>
<th>GROUP</th>
<th>INBOUND</th>
<th>OUTBOUND</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.10.10</td>
<td>up</td>
<td>iotps0</td>
<td>net0</td>
<td>net0 net1</td>
</tr>
<tr>
<td>192.168.10.15</td>
<td>up</td>
<td>iotps0</td>
<td>net1</td>
<td>net0 net1</td>
</tr>
<tr>
<td>192.0.0.100</td>
<td>up</td>
<td>acctg1</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>192.0.0.101</td>
<td>up</td>
<td>acctg1</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>128.0.0.100</td>
<td>up</td>
<td>field2</td>
<td>net2</td>
<td>net2 net7</td>
</tr>
<tr>
<td>128.0.0.101</td>
<td>up</td>
<td>field2</td>
<td>net7</td>
<td>net2 net7</td>
</tr>
<tr>
<td>128.0.0.102</td>
<td>down</td>
<td>field2</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
### ADDRESS
-n オプションが -a オプションと組み合わせて使用された場合は、ホスト名またはデータアドレスを指定します。

### STATE
IPMP インタフェースのアドレスが、up したがって使用可能、down したがって使用不可能、のいずれであるかを示します。

### GROUP
特定のデータアドレスをホストする IPMP IP インタフェースを指定します。

### INBOUND
特定のアドレスのパケットを受信するインタフェースを識別します。このフィールドの情報は、外部のイベントに応じて変わる可能性があります。たとえば、データアドレスが停止している場合や、IPMP グループ内にアクティブな IP インタフェースが1つも残っていない場合、このフィールドは空になります。空のフィールドは、この特定のアドレス宛ての IP パケットをシステムが受け入れていないことを示します。

### OUTBOUND
特定のアドレスを発信元アドレスとして使用したパケットを送信するインタフェースを識別します。INBOUND フィールドと同様に、OUTBOUND フィールドの情報も外部のイベントに応じて変わる可能性があります。空のフィールドは、システムがこの特定の発信元アドレスでパケットを送信していないことを示します。このフィールドが空である場合、それは、アドレスが停止しているからか、あるいはグループ内にアクティブな IP インタフェースが1つも残っていないからです。

### グループのベースとなる IP インタフェースに関する情報取得する方法

この手順は、IPMP グループのベースとなる IP インタフェースに関する情報を表示する場合に使用します。NIC、データリンク、および IP インタフェースの対応関係については、22ページの「Oracle Solaris のネットワークスタック」を参照してください。

#### IPMP のインタフェース情報を表示します。

```
$ ipmpstat -i
```

<table>
<thead>
<tr>
<th>INTERFACE</th>
<th>ACTIVE</th>
<th>GROUP</th>
<th>FLAGS</th>
<th>LINK</th>
<th>PROBE</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>net0</td>
<td>yes</td>
<td>itops0</td>
<td>--mb--</td>
<td>up</td>
<td>ok</td>
<td>ok</td>
</tr>
<tr>
<td>net1</td>
<td>yes</td>
<td>itops0</td>
<td>-------</td>
<td>up</td>
<td>disabled</td>
<td>ok</td>
</tr>
<tr>
<td>net3</td>
<td>no</td>
<td>acctg1</td>
<td>-------</td>
<td>unknown</td>
<td>unknown</td>
<td>offline</td>
</tr>
<tr>
<td>net4</td>
<td>no</td>
<td>acctg1</td>
<td>is-----</td>
<td>down</td>
<td>unknown</td>
<td>failed</td>
</tr>
<tr>
<td>net2</td>
<td>yes</td>
<td>field2</td>
<td>--mb--</td>
<td>unknown</td>
<td>ok</td>
<td>ok</td>
</tr>
<tr>
<td>net6</td>
<td>no</td>
<td>field2</td>
<td>-i-----</td>
<td>up</td>
<td>ok</td>
<td>ok</td>
</tr>
<tr>
<td>net5</td>
<td>no</td>
<td>field2</td>
<td>-------</td>
<td>up</td>
<td>failed</td>
<td>failed</td>
</tr>
<tr>
<td>net7</td>
<td>yes</td>
<td>field2</td>
<td>--mb--</td>
<td>up</td>
<td>ok</td>
<td>ok</td>
</tr>
</tbody>
</table>

### 各 IPMP グループのベースとなる各インタフェースを指定します。
このインタフェースが機能していて使用中 (yes) であるか、それともそうではない (no) かを示します。

IPMP インタフェースの名前を指定します。匿名グループの場合、このフィールドは空になります。匿名グループの詳細については、in.mpathd(1M) のマニュアルページを参照してください。

ベースとなるインタフェースのステータスを示します。これは次のいずれか、あるいはその任意の組み合わせになります。

- i は、このインタフェースに INACTIVE フラグが設定されていることを示します。
- s は、このインタフェースがスタンバイインタフェースとして構成されていることを示します。
- m は、このインタフェースがこの IPMP グループの IPv4 マルチキャストトラフィックの受信をシステムによって指定されていることを示します。
- b は、このインタフェースがこの IPMP グループのプロードキャストトラフィックの受信をシステムによって指定されていることを示します。
- N は、このインタフェースがこの IPMP グループの IPv6 マルチキャストトラフィックの受信をシステムによって指定されていることを示します。
- d は、このインタフェースが停止しており、したがって使用不可能であることを示します。
- h は、このインタフェースが重複する物理ハードウェアアドレスを別のインタフェースと共有しており、オンラインになっていることを示します。h フラグは、このインタフェースが使用不可能であることを示します。

リンクベースの障害検出の状態を示します。これは、次の状態のいずれかになります。

- up または down は、リンクの使用可能または使用不可能を示します。
- unknown は、リンクが up、down のいずれであるかの通知をドライバがサポートしておらず、したがってドライバがリンクの状態変化を検出しないことを示します。

検査用アドレスが構成されたインタフェースについて、次のようにプローべベースの障害検出の状態を指定します。

- ok は、プローべが機能しており、アクティブであることを示します。
IPMP情報の監視

- **failed** は、プロープベースの障害検出が、このインタフェースが動作していないことを検出したことを示します。
- **unknown** は、適切なプロープターゲットが見つからなかったため、プロープを送信できないことを示します。
- **disabled** は、このインタフェースではIPMP検査用アドレスが構成されていないことを示します。したがって、プロープベースの障害検出は無効になっています。

次のように、このインタフェース全体の状態を指定します。

- **ok** は、このインタフェースがオンラインになっている、障害検出手法の構成に基づいて正常に動作していることを示します。
- **failed** は、このインタフェースのリンクが停止しているか、あるいはこのインタフェースはトラフィックを送受信できないとプロープ検出が判定したために、このインタフェースが動作していないことを示します。
- **offline** は、このインタフェースが使用不可能であることを示します。通常、次の状況の下ではインタフェースがオフラインに切り替えられます。
  - インタフェースがテスト中である。
  - 動的再構成が実行中である。
  - このインタフェースが、重複するハードウェアアドレスを別のインタフェースと共用している。
- **unknown** は、プロープベースの障害検出のプロープターゲットが見つからないためにIPMPインタフェースのステータスを判定できないことを示します。

▼ IPMPのプロープターゲット情報を取得する方法

この手順は、IPMPグループ内の各IPインタフェースに関連付けられたプロープターゲットを監視する場合に使用します。

- **IPMPのプロープターゲットを表示します。**

```
$ ipmpstat -nt

INTERFACE MODE TESTADDR TARGETS
net0 routes 192.168.85.30 192.168.85.1 192.168.85.3
net1 disabled -- --
net2 disabled -- --
net3 disabled -- --
net4 routes 192.1.2.200 192.1.2.1
net5 multicast 128.9.0.200 128.0.0.1 128.0.0.2
net6 multicast 128.9.0.201 128.0.0.2 128.0.0.1
net7 multicast 128.9.0.202 128.0.0.1 128.0.0.2
```

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化 - 2011年12月
ipmpstat -nt

<table>
<thead>
<tr>
<th>INTERFACE</th>
<th>MODE</th>
<th>TESTADDR</th>
<th>TARGETS</th>
</tr>
</thead>
<tbody>
<tr>
<td>net3</td>
<td>transitive</td>
<td>&lt;net1&gt;</td>
<td>&lt;net1&gt; &lt;net2&gt; &lt;net3&gt;</td>
</tr>
<tr>
<td>net2</td>
<td>transitive</td>
<td>&lt;net1&gt;</td>
<td>&lt;net1&gt; &lt;net2&gt; &lt;net3&gt;</td>
</tr>
<tr>
<td>net1</td>
<td>routes</td>
<td>172.16.30.100</td>
<td>172.16.30.1</td>
</tr>
</tbody>
</table>

**INTERFACE**

IPMPグループのベースとなるインタフェースを指定します。

**MODE**

- プローブターゲットを取得するための方法を指定します。
- routesは、プローブターゲットの検索にシステムの経路指定テーブルが使用されることを示します。
- mcastは、ターゲットの検索にマルチキャストICMPプローブが使用されることを示します。
- disabledは、このインタフェースでプローブベースの障害検出が無効になっていることを示します。
- transitiveは、2番目の例に示したように、障害検出に推移的プローブが使用されることを示します。推移的プローブと検査用アドレスを同時に使用してプローブベースの障害検出を実装することはできない点に注意してください。検査用アドレスを使用しない場合は、推移的プローブをオンに切り替える必要があります。推移的プローブを使用しない場合は、検査用アドレスを構成する必要があります。概要については、284ページの「検査信号ベースの障害検出」を参照してください。

**TESTADDR**

ホスト名または、-nオプションが-tオプションと組み合わせて使用された場合はプローブの送受信用としてこのインタフェースに割り当てられたIPアドレス、を指定します。

推移的プローブが使用された場合、インタフェースの名前は、データ受信用としてアクティブに使用されていない、ベースとなるIPインタフェースを表します。また、これらの名前は、指定されたこれらのインタフェースの発信元アドレスを使用して推移的テストプローブが送出されていることも示しています。データを受信するアクティブなベースとなるIPインタフェースの場合、表示されるIPアドレスは、送信ICMPプローブの発信元アドレスを示します。

注 - IPインタフェースにIPv4検査用アドレスとIPv6検査用アドレスの両方が設定されている場合、プローブターゲットの情報は各検査用アドレスについて個別に表示されます。

**TARGETS**

現在のプローブターゲットを空白区切りリストとして一覧表示します。-nが-tオプションと組み合わせて使用された場合、プローブターゲットはホスト名、「IPアドレスのいずれかとして表示されます。
▼ IPMPのプローブを監視する方法

この手順は、進行中のプローブを監視する場合に使用します。プローブ監視用のコマンドを発行すると、Ctrl-Cでコマンドを終了するまで、システム上のプローブのアクティビティーに関する情報が継続的に表示されます。このコマンドを実行するには、Primary Administrator特権を持っている必要があります。

1 管理者になります。
詳細は、『Oracle Solarisの管理・セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 進行中のプローブに関する情報を表示します。

```
# ipmpstat -pn
TIME INTERFACE PROBE NETRTT RTT RTTAVG TARGET
0.11s net0 509 0.51ms 0.76ms 0.76ms 192.168.85.1
0.17s net4 612 -- -- -- 192.1.2.1
0.25s net2 602 0.61ms 1.10ms 1.10ms 128.0.0.1
0.26s net6 602 -- -- -- 128.0.0.2
0.25s net5 601 0.62ms 1.20ms 1.10ms 128.0.0.1
0.26s net7 603 0.79ms 1.11ms 1.10ms 128.0.0.1
1.66s net4 613 -- -- -- 192.1.2.1
1.70s net0 603 0.63ms 1.10ms 1.10ms 192.168.85.3
```

```
# ipmpstat -pn
TIME INTERFACE PROBE NETRTT RTT RTTAVG TARGET
1.39s net4 612 1.05ms 1.06ms 1.15ms <net1>
1.39s net1 612 1.00ms 1.42ms 1.48ms 172.16.30.1
```

TIME ipmpstat コマンドが発行された時点を基準にしたプローブの送信時間
を指定します。ipmpstatが開始される前にプローブが起動された場合、
コマンドが発行された時点を基準にした負の値が時間が表示され
ます。

INTERFACE プローブの送信先となるインタフェースを指定します。

PROBE プローブを表す識別子を指定します。障害検出に推移的プローブが使
用される場合は、この識別子に、推移的プローブの場合はtという接
頭辞が、ICMPプローブの場合はiという接頭辞が付きます。

NETRTT プローブの合計ネットワーク往復時間を指定し、ミリ秒で測定され
ます。NETRTTは、IPモジュールがプローブを送信した時点から、IPモ
ジュールがターゲットからackパケットを受け取った時点までの時間
をカバーします。プローブが失われたとin.mpathdデーモンが判定し
た場合、このフィールドは空になります。

RTT プローブの合計往復時間を指定し、ミリ秒で測定されます。RTT
は、デーモンがプローブ送信用コードを実行した時点から、デーモンが
ターゲットからのackパケットの処理を完了した時点までの時間を
カバーします。ブローブが失われたとin.mpathdデーモンが判定した場合、このフィールドは空になります。NRTTに存在しないスパイクがRTTで発生する場合、それは、ローカルシステムが過負荷状態であることを示している可能性があります。

RTTAVG ローカルシステムとターゲット間の、このインタフェース経由でのブローブの平均往復時間を指定します。平均往復時間は、低速なターゲットの特定に役立ちます。データが不十分で平均を計算できない場合、このフィールドは空になります。

TARGET ホスト名または、-nオプションが .p と組み合わせて使用された場合はブローブの送信元となるターゲットアドレス、を指定します。

▼ スクリプト内で ipmpstat コマンドの出力をカスタマイズする方法

ipmpstat を使用する場合はデフォルトで、80列に収まるもっとも意味のあるフィールドが表示されます。出力では、ipmpstat -p構文の場合を除き、ipmpstat コマンドで使用したオプションに固有のフィールドがすべて表示されます。表示対象のフィールドを指定する場合は、-oオプションを、コマンドの出力モードを決定するほかのオプションと組み合わせて使用します。このオプションは特に、コマンドをスクリプトから発行したりコマンドエイリアスを使用して発行したりする場合に便利です。

● 出力をカスタマイズするには、次のいずれかのコマンドを発行します。
  ■ ipmpstat コマンドの選択されたフィールドを表示するには、-oオプションを、特定の出力オプションと組み合わせて使用します。たとえば、グループ出力モードのGROUPNAMEフィールドとSTATEフィールドのみを表示するには、次のように入力します。

    $ ipmpstat -g -o groupname,state

    GROUPNAME | STATE
    --------- | ----
    itops0    | ok   
    accgt1    | failed
    field2    | degraded

  ■ 特定のipmpstatコマンドのフィールドをすべて表示するには、次の構文を使用します。

    # ipmpstat -o all
ipmpstat コマンドのマシンによる解析が可能な出力を生成する方法

ipmpstat -P 構文を使用して、マシンによる解析が可能な情報を生成できます。-P オプションは、特にスクリプト内で使用するためのものです。マシンによる解析が可能な出力は、次のように通常の出力とは異なります。

- ヘッダーは省略されます。
- 各フィールドがコロン(;)で区切られます。
- 空の値を含むフィールドは、三重ダッシュ(---)が設定されるのではなく、空になります。
- 複数のフィールドが要求された場合に、あるフィールドにリテラルのコロン(;)またはバックスラッシュ(\)が含まれていると、それらの文字の接頭辞としてバックスラッシュ(\)を付加することで、それらの文字がエスケープされ除外されます。

ipmpstat -P 構文を正しく使用するには、次の規則に従います。

- -o option fields を -P オプションともに使用する。
- -o all は -P オプションでは決して使用しない。

これらの規則のいずれかを無視した場合は、ipmpstat -P が失敗します。

マシンによる解析が可能な形式でグループ名、障害検出時間、およびベースとなるインターフェースを表示するには、次のように入力します。

```bash
$ ipmpstat -P -o -g groupname,fdt,interfaces
itops0:10.00s:net0 net1
acctg1: [net3 net4] field2:20.00s:net2 net7 (net5) [net6]
```

グループ名、障害検出時間、およびベースとなるインターフェースは、グループ情報フィールドです。したがって、-o -g オプションを -P オプションとともに使用します。

例 15-8 スクリプト内での ipmpstat -P の使用

このサンプルスクリプトは、特定の IPMP グループの障害検出時間を表示します。

```bash
def getfdt() {
    ipmpstat -gP -o group,fdt | while IFS=: read group fdt; do
        [[ "$group" = "$1" ]] && {
            echo "$fdt"; return;
        }
    done
}
```
16

LLDPによるネットワーク接続情報の交換

この章では、システムがリンク層検出プロトコル (LLDP) を使用して、ローカル
ネットワーク全体にわたってシステムおよびネットワーク接続情報を交換できるよ
うにする方法について説明します。

Oracle Solaris での LLDP の概要

LLDP は、トポロジの検出を目的として、ローカルネットワーク全体にわたって情報
を通知するために使用されます。このプロトコルを使用すると、システムは、接続
や管理の情報をネットワーク上のほかのシステムに通知できます。これらの情報に
は、システムの機能、管理アドレス、およびその他の関連情報を含めることができます。
このプロトコルではまた、同じシステムが、同じローカルネットワーク上に
存在するほかのシステムに関する同様の情報を受信することもできます。

Oracle Solaris では、優先順位ベースのフロー制御 (PFC) やアプリケーション TLV など
の DCB 機能に関する構成情報を交換するためのデータセンターブリッジング (DCB)
も LLDP のサポートに含まれています。

LLDP を使用すると、システム管理者は、特に仮想ローカルエリアネットワーク
(VLAN)、リンク集約、その他のリンクタイプを含む複雑なネットワークで、誤った
システム構成を容易に検出できます。

LLDP 実装のコンポーネント

LLDP は、次のコンポーネントを使用して実装されています。

- LLDP 機能を有効にするには、LLDP パッケージがインストールされている必要が
あります。このパッケージは、LLDP デーモン、コマンド行ユーティリ
ティー、サービスマニフェストとスクリプトのほか、LLDP が動作するために必
要なその他のコンポーネントを提供します。
LLDP実装のコンポーネント

- lldp サービスは、svcadm コマンドによって有効になります。このサービスは LLDP デーモンを管理するとともに、このデーモンの起動、停止、再起動、または更新を行います。このサービスは、デフォルトでは無効になっています。そのため、LLDP を使用するには、最初にこのサービスをシステムでグローバルに有効にする必要があります。lldp サービスが有効になり、デーモンが起動されたら、システム管理者によって決定された個々のリンク上で LLDP 機能を有効にすることができます。

- lldpadm コマンドは個々のリンク上の LLDP を管理し、たとえば、LLDP の動作モードを構成したり、送信される TLV (Time-Length-Value) ユニットを指定したり、DCB アプリケーションの情報を構成したりするために使用されます。特に、このコマンドは、エージェントごとの LLDP プロバティーをグローバルな LLDP プロバティーを設定するために使用されます。lldpadm コマンドの一般的なサブコマンドは、dladm およびipadm コマンドの各サブコマンドに対応しています。
  - lldpadm set.* は、指定された LLDP プロバティーに1つ以上の値を設定するアクションの実行を指定します。
  - lldpadm show.* は、指定された LLDP プロバティーに設定されている値を表示します。
  - lldpadm reset.* は、指定された LLDP プロバティーの構成をデフォルト値に戻します。

これらのサブコマンドの使用は、以降の節に示されています。lldpadm コマンドについての詳細は、lldpadm(1M) のマニュアルページを参照してください。

- LLDP ライブラリ (lllldp.so) は、リンク上の LLDP 情報を取得したり、LLDP パケットを解析したり、その他の機能を実行したりするために使用できる API を提供します。

- LLDP エージェントは、LLDP が有効になっている物理 NIC に関連付けられた LLDP インスタンスです。LLDP エージェントは、その関連付けられた NIC 上の LLDP の動作を制御します。LLDP エージェントは、物理 NIC 上でのみ構成できます。

- LLDP デーモン (lldpd) は、システム上の LLDP エージェントの管理プログラムとして機能します。また、SNMP (Simple Network Management Protocol) を経由してシステム上で受信される LLDP 情報を取得するために、SNMP のためのデーモンである snmpd とも相互作用します。さらに、このデーモンは sysevents 情報を送信したり、LLDP ライブラリからのエラーに応答したりします。

次の節では、LLDP エージェントについてさらに詳細に説明します。
LLDP エージェントの機能

LLDP エージェントは、プロトコルデータユニット (PDU) とも呼ばれる LLDP パケットを送信したり、受信したりします。このエージェントは、次の 2 つのタイプのデータストア内で、これらのパケットに含まれている情報を管理および格納します。

- ローカル管理情報ベース (ローカル MIB)。このデータストアには、LLDP エージェントが有効になっている特定のリンクに関連するネットワーク情報が含まれています。ローカル MIB には、一般的な情報を有する情報の両方が含まれます。たとえば、シャーシ ID は、システム上のすべての LLDP エージェントの間で共有されている一般的な情報です。ただし、ポート番号は、システムのデータリンクによって異なります。そのため、各エージェントは、独自のローカル MIB を管理します。
- リモート MIB。このデータストア内の情報は、ローカルネットワーク上のほかのシステムに関連しています。

LLDP エージェントの動作方法の構成

LLDP エージェントは、次のモードで動作するように構成できます。

- 送信のみ (txonly) モードでは、このエージェントは受信 LLDP パケットを処理しません。そのため、リモート MIB は空です。
- 受信のみ (rxonly) モードでは、このエージェントは受信 LLDP パケットのみを処理し、その情報をリモート MIB 内に格納します。ただし、ローカル MIB からの情報は送信されません。
- 送信と受信の両方の (both) モードでは、このエージェントは、LLDP パケットを送信および受信します。両方のタイプの MIB がアクティブに使用されます。このモードではまた、ベースとなるリンクによってサポートされる DCB 機能も自動的に有効になります。
- 無効になった (disable) モードでは、このエージェントは存在しません。

▼ LLDP を有効にする方法

この手順では、LLDP をシステム上ではじめて有効にします。

1 LLDP パッケージをインストールします。

```
# pkg install lldp
```
LLDPエージェントの機能

注 - Oracle Solaris パッケージに関する概要とそのインストール方法については、『Oracle Solaris の管理: 一般的なタスク』の第 12 章「ソフトウェアパッケージの管理(手順)」を参照してください。

2 システム上で LLDP サービスを開始します。
   # svcadm enable svc:/network/lldp:default

3 LLDP を有効にするデータリンクを識別します。

4 そのデータリンクの LLDP エージェントの動作モードを設定します。
   # lldpadm set-agentprop -p mode=value agent
   ここで、value にはいずれかの動作モードを指定することができ、agent には LLDP が有効になっているデータリンクの名前を使用します。

注 - コマンドを使いやすくするために、lldpadm コマンドの各サブコマンドは省略で入力できます。たとえば、lldpadm set-agentprop は、代わりに lldpadm set-ap として入力できます。各サブコマンドとその省略形については、lldpadm(1M)のマニュアルページを参照してください。

5 LLDP エージェントの動作モードを確認するには、次のコマンドを入力します。
   # lldpadm show-agentprop -p mode agent

6 LLDP エージェントを無効にするには、次のコマンドのどちらかを使用します。
   - lldpadm set-agentprop -p mode=disable agent
   - lldpadm reset-agentprop -p mode agent

7 システム全体の LLDP を無効にするには、次のように入力します。
   # svcadm disable svc:/network/lldp:default

例 16-1 複数のデータリンク上の LLDP を有効にする

この例では、システムに 2 つのデータリンク net0 と net1 が存在し、LLDP が LLDP エージェントごとに異なるモードで有効になっています。1 つのエージェントが LLDP パケットの送信と受信の両方によって動作するのに対して、もう一方のエージェントは LLDP パケットの送信のみを行います。

# svcadm enable svc:/network/lldp:default
# lldpadm set-agentprop -p mode=both net0
# lldpadm set-agentprop -p mode=txonly net1
通知する情報内容の構成

LLDPエージェントは、LLDPパケットまたはLLDPDUでシステムおよび接続情報を送信します。このようなパケットには、TLV (Type-Length-Value) 形式で個別にフォーマットされた情報ユニットが含まれています。そのため、これらの情報ユニットはTLVユニットとも呼ばれます。特定のTLVユニットは必須であり、LLDPが有効になったときにデフォルトでLLDPパケットに含まれます。必須のTLVユニットは次のとおりです。

- シャーシID
- ポートID
- TTL (生存期間)
- PDUの終了

シャーシIDがhostidコマンドによって生成された情報であるのに対して、ポートIDは物理NICのMACアドレスです。リンクの数に応じて、1つのシステム内で複数のLLDPエージェントを有効にすることができます。シャーシIDとポートIDの組み合わせによってエージェントが一意に識別され、システム上のほかのエージェントから区別されます。

lldpadmコマンドを使用して、LLDPパケットから必須のTLVユニットのいずれかを除外することはできません。

省略可能なTLVユニットをLLDPパケットに追加することができます。これらの省略可能なTLVユニットは、ベンダーが、通知されるベンダー固有のTLVユニットを挿入する手段になります。TLVユニットは個々の組織—意識別子(OUI)によって識別され、これらのOUIがIEEE 802.1仕様またはIEEE 802.3仕様のどちらであるかに応じて入力されます。TLVのタイプごとの値を設定できるように、各タイプに対応するLLDPエージェントプロパティが作成されます。

次の表は、TLVのタイプまたはグループ、それに対応するプロパティ名、プロパティごとのTLVユニット、およびそれらの説明を示しています。

<table>
<thead>
<tr>
<th>TLVのタイプ</th>
<th>プロパティ名</th>
<th>TLV</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>基本的な管理</td>
<td>basic-tlv</td>
<td>sysname, portdesc, syscapab, sysdesc, mgmtaddr</td>
<td>通知されるシステム名、ポートの説明、システムの機能、システムの説明、および管理アドレスを指定します。</td>
</tr>
</tbody>
</table>

**表16-1** LLDPエージェントに対して有効にできるTLVユニット

第16章・LLDPによるネットワーク接続情報の交換
LLDエージェントの機能

<table>
<thead>
<tr>
<th>TLVのタイプ</th>
<th>プロバティー名</th>
<th>TLV</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.1 OUI</td>
<td>dot1-tlv</td>
<td>vlanname, pvid, linkaggr, pfc, appln</td>
<td>通知されるVLAN名、ポートのVLAN ID、リンク集約、ポートの説明、およびアプリケーションTLVを指定します。</td>
</tr>
<tr>
<td>802.3 OUI</td>
<td>dot3-tlv</td>
<td>max-framesize</td>
<td>通知される最大フレームサイズを指定します。</td>
</tr>
<tr>
<td>Oracle固有的OUI</td>
<td>virt-tlv</td>
<td>vnic</td>
<td>仮想ネットワークが構成されている場合、または、通知されるVNICを指定します。</td>
</tr>
</tbody>
</table>

LLDPが有効になったときにパケットに含まれるTLVユニットを指定するには、これらのプロバティーのいずれかを構成します。

### LLDパケットのTLVユニットを指定する方法

この手順は、LLDパケットで通知されるTLVユニットを追加する方法を示しています。LLDパケットのTLVユニットを設定するには、lldpadm set-agentpropサブコマンドを使用します。

1. 必要な場合は、追加するTLVユニットを含めることのできるLLDエージェントプロバティーを識別します。

   このサブコマンドでは、また、プロバティーごとにすでに設定されているTLVユニットも表示されます。

   ```
   # lldpadm show-agentprop agent
   ```

   プロバティーを指定しない場合は、すべてのLLDエージェントプロバティーとそれらのTLV値が表示されます。

2. プロバティーにTLVユニットを追加します。

   ```
   # lldpadm set-agentprop -p property{+|-}=value[,...] agent
   ```

   複数の値を受け入れるプロバティーには、+|-の修飾子が使用されます。これらの修飾子を使用すると、一覧に値を追加する(+)か、または削除する(-)ができます。これらの修飾子を使用しない場合は、設定する値により、以前にそのプロバティーに対して定義されていたすべての値が置き換えられます。

3. (省略可能) プロバティーの新しい値を表示します。

   ```
   # lldpadm show-agentprop -p property agent
   ```
例 16-2 LLDPパケットへの省略可能なTLVユニットの追加

この例では、LLDP エージェント net0 がすぐに、パケットで VLAN 情報を通知するように構成されています。それに加えて、システムの機能、リンク集約、およびネットワーク仮想化の情報も通知されるようにします。ただし、パケットからVLAN の説明は削除します。

```
# lldpadm show-agentprop net0
# lldpadm set-agentprop -p dot1-tlv+=linkaggr net0
AGENT PROPERTY  PERM VALUE  DEFAULT POSITIVE
    net0 basic-tlv rw sysname sysdesc none    disable none, portdesc, sysname, sysdesc, syscapab, mgmtaddr, all
    net0 dot1-tlv rw vlanname, pvid, pfc none    none, vlanname, pvid, linkaggr, pfc, appln, all
    net0 dot3-tlv rw max-framesize none none, max-framesize, all
    net0 virt-tlv rw none none none, vnics, all
```

```
# lldpadm set-agentprop -p basic-tlv=syscapab, dot1-tlv+=linkaggr, virt-tlv=vnics net0
# lldpadm set-agentprop -p dot1-tlv+=portdesc, dot3-tlv=none net0
# lldpadm show-agentprop -p net0
AGENT PROPERTY  PERM VALUE  DEFAULT POSITIVE
    net0 basic-tlv rw sysname, sysdesc none    disable none, portdesc, sysname, sysdesc, syscapab, mgmtaddr, all
    net0 dot1-tlv rw vlanname, pvid, linkaggr none    none, vlanname, pvid, linkaggr, pfc, appln, all
    net0 dot3-tlv rw max-framesize none none, max-framesize, all
    net0 virt-tlv rw vnics none none none, vnics, all
```

**TLV ユニットの管理**

各 TLV ユニットにはプロパティがあり、これらのプロパティは特定の値を使用してさらに構成できます。その TLV ユニットが LLDP エージェントのプロバティとして有効になると、その TLV ユニットは、指定された値のみを使用してネットワーク内で通知されます。たとえば、システムの機能を通知する TLV 値 syscapab を考えてみます。これらの機能には、ルーター、ブリッジ、リピータ、電話などのデバイスに対するサポートが含まれる可能性があります。ただし、ルーターやブリッジなどの、特定のシステム内で実際にサポートされている機能のみが通知されるように syscapab を設定できます。
TLVを管理するための手順は、グローバルなTLVまたはエージェントごとのTLVのどちらを構成しているかによって異なります。

グローバルなTLVは、システム上のすべてのLLDPエージェントに適用されます。次の表は、グローバルなTLV値とそれに対応する、取り得る構成を示しています。

表16-2 グローバルなTLVとそのプロパティ

<table>
<thead>
<tr>
<th>TLV名</th>
<th>TLVプロパティ名</th>
<th>取り得るプロパティ値</th>
<th>値の説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>syscapab</td>
<td>supported</td>
<td>other、repeater、bridge、wlan-ap、router、telephone、docsis-cd、station、cvlan、sylvan、tpmr</td>
<td>システムの主要なサポートされている機能を表します。デフォルト値は、router、station、およびbridgeです。</td>
</tr>
<tr>
<td>enabled</td>
<td>supported</td>
<td>に対応して示されている値のサブセット</td>
<td>システムの有効になってい る機能を表します。</td>
</tr>
<tr>
<td>mgmtaddr</td>
<td>ipaddr</td>
<td>ipv4またはipv6</td>
<td>ローカルのLLDPエージェントに関連付けられるIPアドレスのタイプを指定します。これらのアドレスは、上位階層エチティーに到達するため に使用され、ネットワーク管理による検出に役立ちま す。指定できるのは1つのタイプだけです。</td>
</tr>
</tbody>
</table>

グローバルな値を取ることのできないTLVユニットは、LLDPエージェントのレベルで管理されます。エージェントごとのTLVユニットでは、指定した値は、特定のLLDPエージェントがそのTLVユニットの転送を有効にしたときに使用されます。

次の表は、LLDPエージェントのTLV値とそれに対応する、取り得る構成を示しています。

表16-3 エージェントごとのTLVユニットとそのプロパティ

<table>
<thead>
<tr>
<th>TLV名</th>
<th>TLVプロパティ名</th>
<th>取り得るプロパティ値</th>
<th>値の説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>pfc</td>
<td>willing</td>
<td>on、off</td>
<td>遠隔マシンからの構成情報を受け入れるか、または拒否するようにLLDPエージェントを設定します。</td>
</tr>
</tbody>
</table>

Oracle Solaris管理:ネットワークインタフェースとネットワーク仮想化・2011年12月
TLVユニットの管理

<table>
<thead>
<tr>
<th>TLV名</th>
<th>TLVプロパティ名</th>
<th>取り得るプロパティ値</th>
<th>値の説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>appln</td>
<td>apt</td>
<td>値は、アプリケーション優先順位表で定義されている情報から取得されます。</td>
<td>アプリケーション優先順位表を構成します。この表には、アプリケーション TLVユニットとそれに対応する優先順位の一覧が含まれています。アプリケーションは、id/selectorのペアで識別されます。この表の内容では、次的形式が使用されます。</td>
</tr>
</tbody>
</table>

次の手順は、グローバルなTLV値を定義する方法を示しています。エージェントごとのTLVユニットを定義する方法については、340ページの「データセンターブリッジング」を参照してください。

▼ グローバルなTLV値を定義する方法

この手順は、特定のTLVユニットにグローバルな値を指定する方法を示しています。グローバルなTLV値を設定するには、llpadm set-tlvprop サブコマンドを使用します。

1. 該当するTLVプロパティを、通知しようとする値を含むように構成します。
参考のため、表16-2を参照してください。
# lldpadm set-tlvprop -p tlv-property=value,[value,value,...] tlv

2. (省略可能)先ほど構成したプロパティの値を表示します。
# lldpadm show-tlvprop

例 16-3 システムの機能と管理IPアドレスの指定
この例では、次の2つの目的を達成します。

- LLDPパケットで通知されるシステムの機能に関する特定の情報を指定します。この目的を達成するには、syscapab TLVユニットの supportedプロパティと enabledプロパティの両方を構成する必要があります。
- 通知で使用される管理IPアドレスを指定します。

# lldpadm set-tlvprop -p supported=bridge,router,repeater syscapab
# lldpadm set-tlvprop -p enabled=router syscapab
# lldpadm set-tlvprop -p ipaddr=192.168.1.2 mgmtaddr
### LLDPアドミンの設定

<table>
<thead>
<tr>
<th>TLVNAME</th>
<th>PROPERTY</th>
<th>PERM</th>
<th>VALUE</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>syscapab</td>
<td>supported</td>
<td>rw</td>
<td>bridge, router, repeater</td>
<td>bridge, router, station</td>
<td>other, router, repeater, bridge, wlan-ap, telephone, docis-cd, station, cvlan, svlan, tpmr bridge, router, repeater</td>
</tr>
<tr>
<td>syscapab</td>
<td>enabled</td>
<td>rw</td>
<td>router</td>
<td>none</td>
<td>--</td>
</tr>
<tr>
<td>mgmtaddr</td>
<td>ipaddr</td>
<td>rw</td>
<td>192.162.1.2</td>
<td>none</td>
<td>--</td>
</tr>
</tbody>
</table>

### データセントラーブリッジング

FCoE (Fibre Channel over Ethernet) パケットをサポートするために、Oracle Solaris での LLDP 実装にはデータセントラーブリッジング (DCB) のサポートが含まれています。

トラフィック交換に従来の Ethernet を使用するネットワークでは、ネットワークがビジー状態のときにパケットが破棄されるリスクが軽微に発生します。FCoE トラフィックに対する主要な要件として、転送中にパケットの破棄が発生する可能性がないことがあります。データセントラーブリッジング交換 (DCBx)、優先順位ベースのフロー制御 (PFC) TLV、およびアプリケーション TLV のサポートにより、パケットの破棄が回避されます。

PFC では、標準の PAUSE フレームが、パケットの優先順位情報を含むように拡張されます。通常、PAUSE フレームは、トラフィック負荷が高い場合に、受信側がすでに受信したパケットを処理できるようにするためにリンク上で送信されます。PFC では、リンク上のすべてのトラフィックを停止するために PAUSE フレームを送信する代わりに、各パケットに対して定義された優先順位に応じてトラフィックが一時停止されます。PFC フレームは、トラフィックを一時停止する必要のある優先順位で送信できます。送信側がその設定の優先順位のトラフィックを停止する一方で、ほかの優先順位のトラフィックは影響を受けません。指定された時間が経過すると、一時停止されたトラフィックを再開できることを通知するために、別の PFC フレームが送信されます。

PFC 構成情報は、DCBx を使用してビアステーション間で交換されます。トラフィック交換におけるビアの PFC 構成が一致している場合、PFC は必要に応じて、トラフィック転送を一時停止または再開することができます。パケットごとに異なる優先順位を割り当てることができるようにするために、アプリケーション TLV を使用して優先順位情報が定義されます。ビアの PFC 構成が一致していない場合は、次の手順に示すように、ほかのビアの構成を受け入れるように PFC TLV をカスタマイズできます。

データセントラーブリッジングは、337 ページの「TLV ユニットの管理」で説明されているエージェントごとの TLV ユニットを構成する方法を示す特定のケースです。
エージェントごとの TLV 値を設定する方法

この手順は、llpadm set-agenttlvprop サブコマンドを使用して LLDP エージェントのレベルで TLV 値を設定する方法を示しています。

1 該当する TLV プロパティを、特定の LLDP エージェントで通知しようとする値を含むように構成します。
   参考のため、表16-3を参照してください。
   # lldpadm set-agenttlvprop -p tlv-property=value[,value,value,...] -a agent tlv-name

2 (省略可能) 先ほど構成したプロパティの値を表示します。
   # lldpadm show-agenttlvprop

例16-4 LLDPエージェントでの情報の受け入れを有効にして、TLVアプリケーションの優先順位を指定する

この例は、pfcやapplnのTLV値をカスタマイズする方法を示しています。この例でのTLVユニットは、FCoEトラフィックに対するDCBの動作方法を指定します。システムは、ローカルの構成がビアの構成に一致していない場合に、ビアのPFC構成を受け入れるように構成されています。この例はまた、LLDPエージェントのアプリケーションTLVに対して優先順位がどのように設定されるかを示しています。

# lldpadm set-agenttlvprop -p willing=on -a net0 pfc
# lldpadm set-agenttlvprop -p apt=8906/1/4 -a net0 appln
# lldpadm show-agenttlvprop

<table>
<thead>
<tr>
<th>AGENT</th>
<th>TLVNAME</th>
<th>PROPERTY</th>
<th>PERM</th>
<th>VALUE</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>net0</td>
<td>pfc</td>
<td>willing</td>
<td>rw</td>
<td>on</td>
<td>off</td>
<td>on,off</td>
</tr>
<tr>
<td>net0</td>
<td>appln</td>
<td>apt</td>
<td>rw</td>
<td>8906/1/4</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

LLDP エージェントの監視

lldpadm show-agent サブコマンドでは、LLDP エージェントによって通知される完全な情報が表示されます。特定のシステムに応じて、通知は、ネットワークのほかの部分に送信される、そのローカルシステムに関する情報である場合があります。あるいは、通知は、そのシステムによって受信される、同じネットワーク上のほかのシステムからの情報である場合があります。

通知を表示する方法

この手順は、LLDP エージェントによって通知されている情報を表示する方法を示しています。情報はローカルまたはリモートのいずれかにできます。ローカル情報は、ローカルシステムから得られます。リモート情報は、ネットワーク上のほかのシステムから得られ、ローカルシステムによって受信されます。
LLDPエージェントの監視

- `lldpdadm show-agent` サブコマンドを適切なオプションとともに使用して、必要な情報を表示します。

- LLDPエージェントによって通知されるローカル情報を表示するには、次のコマンドを入力します。
  
  ```
  # lldpdadm show-agent -l agent
  ```

- LLDPエージェントによって受信されるリモート情報を表示するには、次のコマンドを入力します。
  
  ```
  # lldpdadm show-agent -r agent
  ```

- ローカル情報またはリモート情報のどちらかを詳細に表示するには、次のコマンドを入力します。
  
  ```
  # lldpdadm show-agent -[l|r]v agent
  ```

例16-5 通知されるLLDPエージェント情報の取得

次の例は、LLDPエージェントによってローカルに、またはリモートから通知されている情報を表示する方法を示しています。デフォルトでは、情報は省略形式で表示されます。-vオプションを使用すると、詳細情報を取得できます。

```
# lldpdadm show-agent -l net0
AGENT CHASSISID PORTID
net0 004bb87f 00:14:4f:01:77:5d

# lldpdadm show-agent -lv net0
Agent: net0
  Chassis ID Subtype: Local(7)
  Port ID Subtype: MacAddress(3)
  Port ID: 00:14:4f:01:77:5d
  Port Description: net0
  Time to Live: 81 (seconds)
  System Name: hosta.example.com
  System Description: SunOS 5.11 dcb-clone-x-01-19-11 i86pc
  Supported Capabilities: bridge,router
  Enabled Capabilities: router
  Management Address: 192.168.1.2
  Maximum Frame Size: 3000
  Port VLAN ID: --
  VLAN Name/ID: vlan25/25
  VNIC PortID/VLAN ID: 02:00:20:72:71:31
  Aggregation Information: Capable, Not Aggregated
  PFC Willing: --
  PFC Cap: --
  PFC MBC: --
  PFC Enable: --
  Application(s) (ID/Select/Priority): --
  Information Valid Until: 117 (seconds)
```

```
# lldpdadm show-agent -r net0
AGENT SYSNAME CHASSISID PORTID
net0 hostb 0083b390 00:14:4f:01:77:ab
```
# lldpadm show-agent -rv net0
Agent: net0
  Chassis ID Subtype: Local(7)
  Port ID Subtype: MacAddress(3)
  Port ID: 00:14:4f:01:59:ab
  Port Description: net0
  Time to Live: 121 (seconds)
  System Name: hostb.example.com
  System Description: SunOS 5.11 dcb-clone-x-01-19-11 i86pc
  Supported Capabilities: bridge, router
  Enabled Capabilities: router
  Management Address: 192.168.1.3
  Maximum Frame Size: 3000
  Port VLAN ID: --
  VLAN Name/ID: vlan25/25
  VNIC PortID/VLAN ID: 02:08:20:72:71:31
  Aggregation Information: Capable, Not Aggregated
    PFC Willing: --
    PFC Cap: --
    PFC MBC: --
    PFC Enable: --
  Application(s) (ID/Sel/Pri): --
  Information Valid Until: 117 (seconds)

▼ LLDP 統計情報を表示する方法
ローカルシステムまたはリモートシステムによって通知されている LLDP パケットに関する情報を取得するために、LLDP 統計情報を表示できます。この統計情報は、LLDP パケットの送信および受信を含む重大なイベントを示します。

1 LLDP パケットの送信および受信に関するすべての統計情報を表示するには、次のコマンドを使用します。
   # lldpadm show-agent -s agent

2 選択された統計情報を表示するには、-o オプションを使用します。
   # lldpadm show-agent -s -o field[,field,...]agent
   ここで、field は show-agent -s コマンドの出力にあるいずれかのフィールド名を示します。

例 16-6 LLDP パケット統計情報の表示
この例は、LLDP パケット通知に関する情報を表示する方法を示しています。

   # lldpadm show-agent -s net0
   AGENT IFRAMES IEER IDISCARD OFRAMES OLENERR TLVDISCARD TLVUNRECOG AGEDOUT
net0 9 0 0 14 0 4 5 0
   このコマンド出力では、次の情報が提供されます。
LLDPエージェントの監視

- **AGENT** は、LLDPエージェントの名前を指定します。これは、このLLDPエージェントが有効になっているデークリンクの名前と同じです。
- **IFRAMES、IEER、およびIDISCARD** には、受信されているパケット、エラーを含む受信パケット、および破棄された受信パケットに関する情報が表示されます。
- **OFRAMES と OLENERR** は、送信パケットと、長さのエラーを含むパケットを示します。
- **TLVDISCARD と TLVUNRECOG** には、破棄されたTLVユニットと、認識されないTLVユニットに関する情報が表示されます。
- **AGEOUT** は、タイムアウトしたパケットを示します。

この例は、システムに受信された9フレームのうち、おそらく標準に準拠していないために5つのTLVが認識されないことを示します。この例はまた、ローカルシステムによって14フレームがネットワークに送信されたことも示しています。
パート III

ネットワーク仮想化およびリソース管理
この章では、ネットワーク仮想化およびリソース制御に関連する基本的な概念について説明します。次の項目が含まれています。

- ネットワーク仮想化
- 仮想ネットワークのタイプ
- 仮想マシンとゾーン
- リソース制御（フロー管理を含む）
- ネットワークの可観測性の向上

これらの機能は、フロー制御を管理したり、システムのパフォーマンスを向上させたり、OS仮想化、ユーティリティーコンピューティング、およびサーバー統合を実現するために必要なネットワーク使用率を構成したりするのに役立ちます。

特定のタスクについては、次の章を参照してください。

- 第19章「仮想ネットワークの構成（タスク）」
- 第22章「ネットワークトラフィックとリソース使用状況の監視」
- 第20章「仮想化環境でのリンク保護の使用」
- 第21章「ネットワークリソースの管理」

ネットワーク仮想化と仮想ネットワーク

ネットワーク仮想化とは、ハードウェアのネットワークリソースとソフトウェアのネットワークリソースを1つの管理単位に結合するプロセスのことです。ネットワーク仮想化の目標は、システムとユーザーにネットワークリソースの効率で、制御され、さらにセキュリティー保護された共有を提供することです。

ネットワーク仮想化の最終生成物は、仮想ネットワークです。仮想ネットワークは、大まかに外部と内部の2つのタイプに分類されます。外部仮想ネットワークは、ソフトウェアによって1つのエンティティとして管理される複数のローカルネットワークで構成されます。従来の外部仮想ネットワークの構成単位は、ス
ネットワーク仮想化と仮想ネットワーク

イッチハードウェアとVLANソフトウェア技術ロジです。外部仮想ネットワークの例には、大規模な企業ネットワークやデータセンターが含まれます。

内部仮想ネットワークは、少なくとも1つの仮想ネットワークインタフェース上に構成されている仮想マシンまたはゾーンを使用した1つのシステムで構成されます。これらのコンテナは、同じローカルネットワーク上に存在するかのように互いに通信可能であり、単一ホスト上の仮想ネットワークが提供されます。仮想ネットワークの構成単位は、仮想ネットワークインタフェースカードまたは仮想NIC(VNIC)と仮想スイッチです。Oracle Solarisのネットワーク仮想化は、内部仮想ネットワークのソリューションを提供します。

ネットワークリソースを結合して、内部と外部の両方の仮想ネットワークを構成できます。たとえば、内部仮想ネットワークを含む個々のシステムを、大規模な外部仮想ネットワークの一部であるLANに構成できます。ここで説明されているネットワーク構成には、結合された内部および外部仮想ネットワークの例が含まれています。

内部仮想ネットワークの構成要素

Oracle Solaris上に構築された内部仮想ネットワークには、次の構成要素が含まれています。

- 少なくとも1枚のネットワークインタフェースカード(NIC)。
- ネットワークインタフェースの上に構成された仮想NIC(VNIC)。
- インタフェース上の最初のVNICとして同時に構成された仮想スイッチ。
- VNICの上に構成されたコンテナ(ゾーンや仮想マシンなど)。

次の図は、これらの構成要素と、それらが1つのシステムにどのように結合されているかを示しています。
この図は、1枚のNICを備えた1つのシステムを示しています。このNICには3つのVNICが構成されています。各VNICが1つのゾーンをサポートします。そのため、VNIC1、VNIC2、およびVNIC3上に、それぞれゾーン1、ゾーン2、およびゾーン3が構成されています。これらの3つのVNICが、実質的に1つの仮想スイッチに接続されています。このスイッチは、各VNICと、それらのVNICが構築されている物理NICの間の接続を提供します。物理インタフェースは、システム外部ネットワーク接続を提供します。

あるいは、etherstubに基づいて仮想ネットワークを作成できます。etherstubは純粋にソフトウェアであるため、仮想ネットワークのための基礎としてネットワークインタフェースを必要としません。

VNICは、物理インタフェースと同じデータリンクインタフェースを備えた仮想ネットワークデバイスです。VNICは、物理インタフェースの上に構成します。VNICをサポートする物理インタフェースの現在の一覧については、Network Virtualization and Resource Control FAQ(http://hub.opensolaris.org/bin/view/Project+crossbow/faq)を参照してください。1つの物理インターフェース上に最大900
ネットワーク仮想化と仮想ネットワーク

のVNICを構成できます。構成されたVNICは、物理NICのように動作します。さらに、システムのリソースは、VNICを物理NICであるかのように扱います。

各VNICは、物理インタフェースに対応する仮想スイッチに暗黙に接続されています。この仮想スイッチは、スイッチのポートに接続されたシステムにスイッチハードウェアによって提供される仮想ネットワーク上のVNIC間に同じ接続を提供します。

Ethernet設計に従うと、スイッチポートが、そのポートに接続されたホストから送信パケットを受信した場合、そのパケットは同1ポート上の宛先に到達できません。この設計は、ゾーンまたは仮想マシンが構成されたシステムでの欠点です。ネットワーク仮想化を使用しない場1面は、期待するルータや仮想マシンまたはゾーンからの送信パケットを、同じシステム上の別の仮想マシンまたはゾーンに通すことはできません。それらの送信パケットは、スイッチポートを経由して外部ネットワークに転送されます。受信パケットは、送信されたのと同じポートを通じて戻ることができないため、宛先のゾーンまたは仮想マシンに到達できません。そのため、同じシステム上の仮想マシンやゾーン間の通信が必要な場合は、ローカルマシン上でこれらのコンテナ間のデータパスが開かれている必要があります。仮想スイッチは、これらのコンテナに、パケットを通すための方法を提供します。

仮想ネットワークを通じたデータの転送方法

図17-1は、1つのシステム上の仮想ネットワークのための単純なVNIC構成を示しています。

仮想ネットワークが構成されている場合、ゾーンは、仮想ネットワークが存在しないシステムと同じ方法で外部のホストにトラフィックを送信します。トラフィックは、そのゾーンからVNICを通じて仮想スイッチに、さらに物理インタフェースへと転送され、その物理インタフェースによってデータがネットワークに送信されます。

しかし、前に説明したEthernetの制限が存在するとしたら、仮想ネットワーク上のあるゾーンがパケットを仮想ネットワーク上の別のゾーンに送信しようとする場合にどうなるでしょうか。図17-1に示すように、ゾーン1がトラフィックをゾーン3に送信する必要があるとします。この場合、パケットはゾーン1から専用のVNIC1を通じて転送されます。そのトラフィックは次に、仮想スイッチを通じてVNIC3に転送されます。次に、VNIC3がそのトラフィックをゾーン3に通します。このトラフィックがシステムを離れることはないため、Ethernetの制限には違反しません。

仮想ネットワークを実装すべきユーザー

OracleのSunサーバー上のリソースを統合する必要がある場合は、VNICと仮想ネットワークの実装を検討してください。ISP、電気通信会社、および大規模な金融機関の統合担当者は、次のネットワーク仮想化機能を使用して、所有するサーバーとネットワークのパフォーマンスを向上させることができます。
リソース制御について

リソース制御とは、システムのリソースを制御された方法で割り当てるプロセスのことです。Oracle Solaris のリソース制御機能を使用すると、システムの仮想ネットワーク上の VNIC の間で帯域幅を共有できます。また、リソース制御機能を使用して、VNIC 仮想マシンの存在しない物理インタフェース上の帯域幅を割り当てたり、管理したりすることもできます。この節では、リソース制御の主な機能を説明したあと、これらの機能のしくみについて簡単に説明します。

帯域幅の管理とフロー制御のしくみ

Searchnetworking.com (http://searchnetworking.techtarget.com) では、帯域幅を「特定の時間（通常は1秒）内にあるボイントから別のボイントに転送できるデータの量」として定義しています。帯域幅の管理を使用すると、物理 NIC の使用可能な帯域幅の一部をコンシューマー（アプリケーションや顧客など）に割り当てることができます。帯域幅は、アプリケーションごと、ポートごと、プロトコルごと、およびアドレスごとに制御できます。帯域幅の管理によって、新しいGLDv3 ネットワークインタフェースから使用可能な帯域幅の効率的な使用が保証されます。

リソース制御機能を使用すると、インタフェースの使用可能な帯域幅に対する一致の制御を実装できます。たとえば、インタフェースの帯域幅を特定のコンシューマーに割り当てることの保証を設定できます。その保証は、アプリケーションまたは企業への割り当てが保証された帯域幅の最小量です。この帯域幅の割り当てられた部分は、共有と呼ばれています。保証を設定することによって、特定の帯域帯域幅がないために機能できないアプリケーションに十分な帯域幅を割り当てることができます。たとえば、ストリーミングメディアや Voice over IP は大量の帯域幅を消費します。リソース制御機能を使用すると、これらの2つのアプリケーションに、正常に動作するための十分な帯域幅を割り当てられることを保証できます。

共有に対して制限を設定することもできます。この制限は、共有が消費できる帯域幅の最大の割り当てです。制限を使用すると、重要でないサービスを抑制して重要なサービスから帯域幅を奪うことがないようにすることができます。
リソース制御について

最後に、コンシューマに割り当てられるさまざまな共有に優先順位を付けることができます。クラスタのハートビートパケットなどの重要なトラフィックにはもっとも高い優先順位を与える一方で、重要性の低いアプリケーションにはより低い優先順位を与えることができます。

たとえば、アプリケーションサービスプロバイダ (ASP) は、顧客が購入する帯域幅共有に基づいた料金ベースのサービスレベルを顧客に提供できます。それにより、サービスレベル契約 (SLA) の一部として、購入された制限を超えない帯域幅の量が各共有に保証されます。サービスレベル契約についての詳細は、「Oracle Solarisの管理: IP サービス」の「サービスレベル契約の実装」を参照してください。優先順位の制御は、SLA の各層、または SLA の顧客によって支払われた各価格に基づいて行われる場合があります。

帯域幅の使用は、フローの管理を通じて制御されます。フローとは、すべてに特定の特性 (ポート番号や宛先アドレスなど) が含まれるパケットのストリームのことです。これらのフローは、トランスポート、サービス、または仮想マシン (ゾーンを含む) によって管理されます。フローは、アプリケーションまたは顧客の購入した共有に対して保証された帯域幅の量を超えることはできません。

VNIC またはフローに保証が割り当てられている場合は、インタフェースがほかのフローまたは VNIC でも使用されていたとしても、その VNIC は指定された帯域幅を保証されます。ただし、割り当てられた保証が機能するのは、それが物理インタフェースの最大帯域幅を超えない場合だけです。

ネットワーク上のリソースの割り当て制御と帯域幅の管理

次の図は、リソース制御を使用してさまざまなアプリケーションを管理する企業ネットワークを示しています。
この図は、リソース制御を使用してネットワークの効率とパフォーマンスを向上させる標準的なネットワークトポロジを示しています。このネットワークでは、VNICやコンテナ（仮想ボリュームや仮想マシンなど）が実装されていません。ただし、統合やその他の目的のために、このネットワーク上でVNICとコンテナを使用できます。

このネットワークは、次の4つの層に分けられています。

- 層0は非武装ゾーン（DMZ）です。これは、外部の世界との間のアクセスを制御するための小さなローカルネットワークです。DMZのシステム上では、リソース制御は使用されません。
- 層1はWeb層であり、2つのシステムが含まれています。最初のシステムは、フィルタリングを実行するブロキシサーバーです。このサーバーには、bge0とbge1の2つのインターフェースがあります。bge0リンクは、ブロキシサーバーを
ネットワーク仮想化およびリソース制御のための可観測性機能

層0上のDMZに接続します。bge0リンクはまた、プロキシサーバーを2番目のシステムであるWebサーバーにも接続します。httpサービスとhttpsサービスは、Webサーバーの帯域幅をほかの標準アプリケーションと共用します。Webサーバーのサイドとその重要な性質のために、httpとhttpsの共有には保証と優先順位付けが必要です。

- 層2はアプリケーション層であり、ここにも2つのシステムが含まれています。プロキシサーバーの2番目のインタフェースであるbge1は、Web層アプリケーション層の間の接続を提供します。アプリケーションサーバーは、スイッチを通じてプロキシサーバー上のbge1に接続します。アプリケーションサーバーには、実行されているさまざまなアプリケーションに与えられる帯域幅の共有を管理するためのリソース制御が必要です。多くの帯域幅が必要とする重要なアプリケーションには、より小規模な、重要性の低いアプリケーションより高い保証と優先順位を与える必要があります。

- 層3はデータベース層です。この層にある2つのシステムは、スイッチを通じてプロキシサーバーのbge1インタフェースに接続します。最初のシステムであるデータベースサーバーは、保証を発行したり、データベースの検索に関連するさまざまなプロセスに優先順位を付けたりする必要があります。2番目のシステムは、ネットワークのバックアップサーバーです。このシステムは、バックアップ中に大量の帯域幅を消費します。ただし、バックアップ動作は通常、夜間に実行されます。リソース制御を使用すると、バックアッププロセスにいつ、もともと高い帯域幅保証をもっとも高い優先順位を与えるかを制御できます。

リソース制御機能を実装すべきユーザー

システムの労率とパフォーマンスを向上させたいと考えているシステム管理者はすべて、リソース制御機能の実装を検討する必要があります。統合担当者は、大規模なサーバーの負荷分散に役立つために、帯域幅共有をVNICEと組み合わせて委任できます。サーバー管理者は、共有の割り当て機能を使用して（たとえば、ASPによって提供される）SLAを実装できます。従来のシステム管理者は、帯域幅の管理機能を使用して特定のアプリケーションを分離したり、優先順位付けしたりすることができます。その後、共有の割り当てによって、個々のコンシューマによる帯域幅の使用を監視することが容易になります。

ネットワーク仮想化およびリソース制御のための可観測性機能

ネットワーク仮想化およびリソース制御には、VNICEやフローなどの制御を設定する前にリソース使用状況を表示するのに役立つ可観測性機能が含まれています。リソース制御の可観測性機能を使用すると、Oracle Solarisの拡張アカウンティングと連携して、システムの統計情報をログに累積できます。ネットワーク仮想化およびリソース制御の可観測性機能には、次ものが含まれます。
ネットワーク仮想化およびリソース制御のための可観測性機能

- 実行中のシステムを監視する機能
- 統計情報をログに記録して報告する機能
- 履歴データをログに記録する拡張アカウンティング機能

新しいflowadmコマンドと、dladmおよびnetstatコマンドの拡張機能によって、ネットワーク仮想化可観測性機能が実装されます。これらのコマンドを使用すると、現在のシステムの使用状況を監視したり、統計データをログに収集したりすることができます。

履歴ログを解析することによって、次のことを作成できます。

- おそらく新世代のネットワークインタフェースから得られる、より多くの帯域幅を使用して、ネットワークリソースを多数のシステムから1つのシステムに統合できるか。これは、VNICと仮想マシンまたは排他的ゾーンを設定する前に行います。

- どのアプリケーションがもっとも多くの帯域幅を消費しているか。この情報は、帯域幅の管理を設定して重要なアプリケーションに特定の時間帯でもっとも多くの帯域幅が保証されるようにするのに役立ちます。たとえば、1日に20時間、ビデオストリームにインタフェースの帯域幅の最大値を保証できます。1日の中の指定された4時間、システムのバックアッププログラムにもっとも高い優先順位を与えることができます。これは、帯域幅の管理を実装する作業の一部として行います。

- 使用された帯域幅に対して顧客にどれだけの金額を請求するか。アプリケーションサービスプロバイダやシステムスペースを貸し出しているその他の企業は、リソース制御の可観測性機能を使用して、料金を支払っている顧客による使用状況を設定できます。一部の企業は顧客にサービスレベル契約を提供しており、そこで顧客は、帯域幅の保証された割合をプロバイダから購入します。可観測性機能を使用すると、顧客がどれだけの帯域幅を使用しているかを表示し、見込まれる超過分に対して請求することができます。その他の企業は、顧客に帯域幅を使用量に基づいて提供します。ここでは、可観測性機能が直接、請求に役立ちます。これは、システム上にリソース制御および（おそらく）VNICと仮想マシンを実装したあとに行います。

次の章第18章「ネットワーク仮想化およびリソース制御の計画」には、統合とリソース制御を計画するために可観測性機能がどこで使用されるかを示すシナリオが含まれています。
第18章
ネットワーク仮想化およびリソース制御の計画

この章には、サイトのネットワーク仮想化およびリソース制御ソリューションをまず評価して、次に設計するのに役立つ情報とシナリオの例が含まれています。この章では、次のシナリオについて説明します。

- 358ページの「1つのシステム上の基本的な仮想ネットワーク」
- 360ページの「1つのシステム上のプライベート仮想ネットワーク」
- 364ページの「従来のネットワークに対するインフラベースのリソース制御」

各シナリオには、特定のシナリオから最大の利点が得られるネットワークのタイプを説明した「最適な使用法」の提案が含まれています。

ネットワーク仮想化およびリソース制御のタスクマップ

次の表は、仮想ネットワークを構成し、そのネットワーク上にリソース制御を実装するためのタスクを示しています。

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>単一ホスト上の仮想ネットワークを設計および計画します。</td>
<td>ローカルネットワークによって提供されるネットワークサービスとアプリケーションを単一ホストに統合します。 このシナリオは、統合担当者やサービスプロバイダに特に役立ちます。</td>
<td>358ページの「仮想ネットワークの計画および設計」</td>
</tr>
</tbody>
</table>
仮想ネットワークの計画および設計

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>単一ホスト上のプライベート仮想ネットワークを設計および計画します。</td>
<td>パブリックアクセスを許可しない仮想ネットワークを実行します。</td>
<td>360 ページの「1つのシステム上のプライベート仮想ネットワーク」</td>
</tr>
<tr>
<td>システムに対する帯域幅の管理とリソース制御をインタフェースごとに提供します。</td>
<td>パケットトラフィックのための特定の量のインタフェース帯域幅を分離し、優先順位を付けて、割り当てます。</td>
<td>364 ページの「従来のネットワークに対するインタフェースベースのリソース制御」</td>
</tr>
</tbody>
</table>

仮想ネットワークの計画および設計

この節では、仮想ネットワークを構成するための2つの異なるシナリオについて説明します。これらのシナリオを調べて、どれがサイトのニーズにもっとも厳密に適合するかの判定に役立ててください。次に、そのシナリオを、特定の仮想化ソリューションを設計するための基礎として使用します。これらのシナリオには、次のものが含まれています。

- 2つのゾーンの基本的な仮想ネットワーク。ネットワークサービスをローカルネットワークから単一ホストに統合するのに特に役立ちます。
- プライベート仮想ネットワーク。アプリケーションやサービスをパブリックネットワークから分離する開発環境に役立ちます。

1つのシステム上の基本的な仮想ネットワーク

図 18–1 は、370 ページの「Oracle Solaris でのネットワーク仮想化のコンポーネントの構成」の節全体にわたる例で使用されている基本的な仮想ネットワーク（つまり、「ボックス内のネットワーク」）を示しています。
図18-1 単一ホスト上の仮想ネットワーク

システム

大域ゾーン

zone1
192.168.3.20

zone2
192.168.3.22

vnic1
192.168.3.20

vnic2
192.168.3.22

仮想スイッチ

e1000g0
192.168.3.70

ローカルネットワーク 192.168.3.0

この仮想ネットワークは、次のもので構成されています。

- 1つのGLDv3ネットワークインタフェース e1000g0。このインタフェースは、パブリックネットワーク192.168.3.0/24に接続されています。インタフェース e1000g0 にはIP アドレス 192.168.3.70 が割り当てられています。
- 最初のVNICを作成したときに自動的に構成される仮想スイッチ。
- 2つのVNIC。vnic1にはIPアドレス 192.168.3.20が割り当てられ、vnic2にはIPアドレス 192.168.3.22が割り当てられています。
- VNICが割り当てられている2つの排他的IPゾーン。vnic1はzone1に割り当てられ、vnic2はzone2に割り当てられています。

この構成のVNICとゾーンによって、パブリックアクセスが許可されます。そのため、これらのゾーンは、e1000g0インタフェースを超えてトラフィックを通すことができます。同様に、外部ネットワーク上のユーザーは、これらのゾーンによって提供されるアプリケーションやサービスに到達できます。

基本的な仮想ネットワークの最適な使用法

「ボックス内のネットワーク」のシナリオを使用すると、プロセスやアプリケーションを単一ホスト上の個々の仮想マシンまたはゾーンに分離できます。さらに、このシナリオは、それぞれが完全に切り離された一連のアプリケーションを実行できる多数のコンテナを含むように拡張できます。このシナリオによってシステム
仮想ネットワークの計画および設計

仮想ネットワークの効率が向上するだけでなく、ローカルネットワークの効率も向上します。そのため、このシナリオは次のユーザーに最適です。

- LANのサービスを1つのシステムに統合したいと考えているネットワーク統合担当者やその他のユーザー。
- 顧客にサービスを貸し出しているすべてのサイト。個々のゾーンまたは仮想マシンを貸し出し、トラフィックを監視して、仮想ネットワーク内の各ゾーンに関するパフォーマンスの測定または請求の目的で統計情報を取得することができます。
- システムの効率を向上させるために、プロセスやアプリケーションを個別のコンテナに分離したいと考えているすべての管理者。

1つのシステム上のプライベート仮想ネットワーク

図18-2は、ネットワークアドレス変換 (NAT) を実行するパケットフィルタリングソフトウェアの背後にプライベートネットワークを含む1つのシステムを示しています。この図は、例19-5で構築されるシナリオを示しています。
図18-2 単一ホスト上のプライベート仮想ネットワーク

このトポロジーは、バブリックネットワーク（ファイアウォールを含む）と、etherstub擬似インタフェース上に構築されたプライベートネットワークを含む1つのシステムを備えています。バブリックネットワークは大域ゾーンで実行され、次の要素で構成されています。

- IPアドレス192.168.3.70が割り当てられたGLDv3ネットワークインタフェースe1000g0。
- IPフィルタソフトウェアに実装されたファイアウォール。IPフィルタの概要については、『Oracle Solarisの管理: IPサービス』の「IPフィルタとは」を参照してください。
- 仮想ネットワークトポロジーの構築に使用される擬似インタフェースであるetherstub0。etherstubは、ホスト上に仮想ネットワークを作成するための機能を提供します。そのネットワークは、外部ネットワークから完全に切り離されていきます。

プライベートネットワークは、次の要素で構成されています。
- プライベートネットワークのVNIC間のパケット転送を提供する仮想スイッチ。
ネットワークリソースに対する制御の実装

- 大域ゾーンのVNICであるvnic0。IPアドレス192.168.0.250が割り当てられています。
- IPアドレス192.168.0.200が割り当てられたvnic1と、IPアドレス192.168.0.220が割り当てられたvnic2。3つのVNICはすべて、etherstub0に構成されています。
- vnictはzone1に割り当てられ、vnict2はzone2に割り当てられています。

プライベート仮想ネットワークの最適な使用法
開発環境で使用されるホストのためのプライベート仮想ネットワークを作成することを検討してください。etherstubフレームワークを使用すると、開発中のソフトウェアまたは機能をプライベートネットワークのコンテナに完全に分離できます。さらに、ファイアウォールソフトウェアを使用して、プライベートネットワークのコンテナから送信された送信バケットのネットワークアドレス変換も行うことができます。このプライベートネットワークは、最終的な配備環境のより小さなバージョンです。

参照先
- 仮想ネットワークを構成し、この章で説明されているこれらのシナリオを実装する手順については、385ページの「プライベート仮想ネットワークの作成」を参照してください。
- VNICと仮想ネットワークに関する概念的な情報については、347ページの「ネットワーク仮想化と仮想ネットワーク」を参照してください。
- ゾーンに関する概念的な情報については、『Oracle Solarisのシステム管理(Oracle Solarisゾーン、Oracle Solaris 10ゾーン、およびリソース管理)』の第15章「Oracle Solarisゾーンの紹介」を参照してください。
- IPフィルタについては、『Oracle Solarisの管理: IPサービス』の"IPフィルタとは"を参照してください。

ネットワークリソースに対する制御の実装
ネットワーク仮想化を使用すると、「ボックス内のネットワーク」を構成することによって、ネットワークの設定をより低いコストで、より効率的に実装できます。また、効率を向上させるために、リソースがネットワークプロセスによってどのように使用されているかを特定するための制御を実装することもできます。ネットワークリソース(リングやCPUなど)に明確に関連したリンクプロパティを作り、ネットワークパケットを処理するようにカスタマイズできます。さらに、ネットワーク使用を管理するためのフローを作成することもできます。ネットワークリソースの制御については、第21章「ネットワークリソースの管理」で詳細に説明されています。

Oracle Solaris 管理: ネットワークインターフェースとネットワーク仮想化・2011年12月
図18-3 は、ブロキシサーバー上的帯域幅を管理する必要のある小規模企業のためのネットワークポリシを示しています。このブロキシサーバーは、パブリックWebサイトだけでなく、そのサイトの内部ネットワーク上のさまざまなサーバーからのサービスが必要な内部クライアントのためのブロキシを提供します。

注- このシナリオには、仮想ネットワークに対するフロー制御を構成する方法は示されておらず、その結果VNICは含まれていません。仮想ネットワーク上のフロー制御については、「仮想ネットワークに対するフロー制御」を参照してください。

図18-3 従来のネットワーク上のブロキシサーバーに対するリソース制御

この図は、ある企業に、非武装ゾーン(DMZ)としても機能するパブリックネットワーク10.10.6.0/8が存在することを示しています。DMZ上のシステムが、IPフィルタファイアウォールを通して、名前からアドレスへの変換(NAT)を提供します。この企業には、ブロキシサーバーとして機能する大規模なシステムが存在します。このシステムには、2つの有線インタフェースと、0から16までの16IDを持つ16のプロセッサセットがあります。このシステムは、IPアドレス10.10.6.5が割り当て
されたインタフェース nge0 を経由してパブリックネットワークに接続されています。このインタフェースのリンク名は DMZ0 です。プロキシサーバーは、DMZ0 を通じて、企業のパブリック Web サイトを経由した HTTP および HTTPS サービスを提供します。

この図にはまた、企業の内部ネットワーク 10.10.12.0/24 も示されています。プロキシサーバーは、IP アドレス 10.10.12.42 が割り当てられたインタフェース nge1 を経由して内部の 10.10.12.0/8 ネットワークに接続されています。このインタフェースのリンク名は internal0 です。プロキシサーバーは、internal0 データリンクを通じて、アプリケーションサーバー 10.10.12.45、データベースサーバー 10.10.12.46、およびバックアップサーバー 10.10.12.47 のサービスを要求する内部クライアントの代わりに動作します。

従来のネットワークに対するインタフェースベースのリソース制御

従来のネットワークに対するインタフェースベースのリソース制御の最適な使用法

頻繁に使用されるシステム、特に、大量の使用可能な帯域幅を提供するより新しい GLDv3 インタフェースを備えたシステムに対するフロー制御を確立することを検討してください。インタフェースベースのフロー制御により、インタフェース、システム、および場合によってはネットワークの効率が向上します。どのタイプのネットワーク上のどのシステムに対してもフロー制御を適用できます。さらに、ネットワークの効率を向上させることが目標である場合は、さまざまなサービスを個々のフローに分離できます。この操作によって、個別のハードウェアおよびソフトウェアリソースが個々のフローに割り当てられるため、これらのリソースが特定のシステム上の各サービスから分離されます。フローを確立したあと、フローごとのトラフィックを監視し、統計情報を収集することができます。そのあと、インタフェース上の使用を制御するために、帯域幅の量と優先順位を割り当てることができます。

参照先

- フロー制御を実装するためのタスクについては、第 21 章「ネットワークリソースの管理」を参照してください。
- 帯域幅の管理とリソース制御に関する概念的な情報については、351 ページの「リソース制御について」を参照してください。
- 詳細な技術的情報については、dladm(1M) と flowadm(1M) のマニュアルページを参照してください。
仮想ネットワークに対するフロー制御

このシナリオは、358 ページの「1つのシステム上の基本的な仮想ネットワーク」で紹介された基本的な仮想ネットワークなどの仮想ネットワーク内でフロー制御を使用する方法を示しています。

図18-4 フロー制御が適用された基本的な仮想ネットワーク

このトポロジーについては、358 ページの「1つのシステム上の基本的な仮想ネットワーク」で説明されています。このホストには、vnic1 と vnic2 の 2 つの VNIC を含む 1 つのネットワークインタフェース e1000g0 があります。zone1 は vnic1 上に構成されており、zone2 は vnic2 上に構成されています。仮想ネットワークに対するリソース管理を行うには、VNIC ごとにフローを作成する必要があります。これらのフローによって、送信側のポートのポート番号や IP アドレスなどの同様の特性を持つパケットが定義され、分離されます。システムの使用ポリシーに基づいて帯域幅を割り当てます。

VNIC トライフィックに対するフロー制御の別の非常に一般的な使用法として、ゾーンを貸し出している企業によるものがあります。顧客向けの各種のサービスレベル契約を作成し、保証された帯域幅の量でゾーンを貸し出します。ゾーンごとにフローを作成すると、各顧客のトラフィックを分離して監視したり、帯域幅の使用状況を監視したりすることができます。サービスレベル契約が厳密に使用量に基づいている場合は、統計情報とアカウンティング機能を使用して顧客に請求できます。
ネットワークリソースに対する制御の実装

フロー制御は、ゾーン経由のトラフィックに対する帯域幅の管理が必要なすべてのネットワークに有効です。アプリケーションサービスプロバイダ(ASP)やインターネットサービスプロバイダ(ISP)などのより大規模な組織は、データセンターやマルチプロセッサシステムのVNICに対するリソース制御を利用できます。個々のゾーンを各種のサービスレベルで顧客に貸し出すことができます。そのため、zone1を標準価格で貸し出し、標準の帯域幅を提供できます。次に、zone2をプレミアム価格で貸し出し、その顧客に高いレベルの帯域幅を提供できます。

▼仮想ネットワーク上のアプリケーションに対する使用ポリシーを作成する方法

1 ホスト上で実行するアプリケーションを一覧表示します。

2 どのアプリケーションがこれまでにもっとも多くの帯域幅を使用したか、またはもっとも多くの帯域幅を必要としてきたかを判定します。たとえば、telnetアプリケーションはシステム上で膨大な量の帯域幅を消費していなくても、頻繁に使用される可能性があります。逆に、データベースアプリケーションは膨大な量の帯域幅を消費しますが、散発的にしか使用されない可能性があります。ゾーンに割り当てる前に、これらのアプリケーションのトラフィックを監視することを検討してください。429ページの「リンク上のネットワークトラフィックに関する統計情報の収集」の説明に従って、dladm show-link コマンドの統計オプションを使用して統計情報を収集できます。

3 これらのアプリケーションを個別のゾーンに割り当てます。

4 トラフィックを分離して制御しようとしている、zone1で実行されているすべてのアプリケーションに対してフローを作成します。

5 サイトに適用されている使用ポリシーに基づいて、フローに帯域幅を割り当てます。

▼仮想ネットワークのためのサービスレベル契約を作成する方法

1 各種のサービスレベルを異なる価格で提供するためのポリシーを設計します。たとえば、基本的なサービスレベル、優れたサービスレベル、および高いサービスレベルを作成し、それに応じて各レベルの価格を設定します。
ネットワークリソースに対する制御の実装

2 顧客に料金を月単位で、サービスレベルごとに請求するか、または実際に消費された帯域幅に基づいて請求するかを決定します。
後者の価格構造を選択した場合は、各顧客の使用量に関する統計情報を収集する必要があります。

3 ホスト上に仮想ネットワークと顧客ごとのコンテナを作成します。
非常に一般的な実装として、各顧客に VNIC 上で実行される独自のゾーンを提供する方法があります。

4 ゾーンごとのトラフィックを分離するフローを作成します。
ゾーンのすべてのトラフィックを分離するには、そのゾーンの VNIC に割り当てられた IP アドレスを使用します。

5 各 VNIC のゾーンに割り当てられた顧客が購入したサービスレベルに基づいて、その VNIC に帯域幅を割り当てます。
仮想ネットワークの構成 (タスク)

この章には、内部仮想ネットワーク、つまり、「ボックス内のネットワーク」を構成するためのタスクが含まれています。次の内容について説明します。

- 369 ページの「仮想ネットワークのタスクマップ」
- 370 ページの「Oracle Solarisでのネットワーク仮想化的コンポーネントの構成」
- 375 ページの「VNIC とゾーンの操作」

仮想ネットワークのタスクマップ

この表は、仮想ネットワークを構成するためのタスク (特定のタスクへのリンクを含む) を示しています。実際の仮想ネットワークのシナリオにすべてのタスクが適用されるわけではないことに注意してください。

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>システム内に VNIC を作成します。</td>
<td>1つ以上の仮想ネットワークインタフェース (VNIC) を作成します。 VNIC は、仮想ネットワークを構築するときの基になる擬似インタフェースです。</td>
<td>371 ページの「仮想ネットワークインタフェースを作成する方法」</td>
</tr>
<tr>
<td>システム内に etherstub を作成します。</td>
<td>1つ以上の etherstub を作成します。 etherstub は、より大規模なネットワークから分離されたプライベート仮想ネットワークを作成できるようにするための仮想スイッチです。</td>
<td>373 ページの「etherstub を作成する方法」</td>
</tr>
<tr>
<td>タスク</td>
<td>説明</td>
<td>手順</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>VNICを使用するためのゾーンを作成します。</td>
<td>VNICと新しいゾーンを作成し、これらを基本的な仮想ネットワークを作成するように構成します。</td>
<td>375ページの「VNICとともに使用するための新しいゾーンの作成」</td>
</tr>
<tr>
<td>ゾーンを VNICを使用するように変更します。</td>
<td>既存のゾーンを仮想ネットワークになるように変更します。</td>
<td>380ページの「既存のゾーンの構成を VNICを使用するように変更する」</td>
</tr>
<tr>
<td>プライベート仮想ネットワークを作成します。</td>
<td>etherstub と VNICを使用して、より大规模なネットワークから分離されたプライベートネットワークを構成します。</td>
<td>385ページの「プライベート仮想ネットワークの作成」</td>
</tr>
<tr>
<td>VNICを削除します。</td>
<td>ゾーン自体を削除することなく、そのゾーンを割り当てられた VNICを削除します。</td>
<td>387ページの「ゾーンを削除することなく仮想ネットワークを削除する方法」</td>
</tr>
</tbody>
</table>

**Oracle Solaris でのネットワーク仮想化のコンポーネントの構成**

この節には、Oracle Solaris でネットワーク仮想化の構成単位を構成するためのタスクが含まれています。基本的なコンポーネントは次のもので構成されます。

- 仮想ネットワークインタフェースカード (VNIC)
- etherstub

VNICは、データリンクの上に作成する擬似インタフェースです。VNICには、自動的に生成された MAC アドレスがあります。使用されているネットワークインタフェースによっては、dladm(1M) のマニュアルページで説明されているように、VNICにデフォルトのアドレス以外の MAC アドレスを明示的に割り当てることができます。VNICは、データリンクにいくつでも必要なだけ作成できます。

etherstubは、システム管理者によって管理される擬似 Ethernet NIC です。VNICは物理リンク上にではなく、etherstub上に作成できます。etherstub 上の VNICは、システム内の物理 NICから独立します。etherstubを使用すると、システム内のかの仮想ネットワークと外部ネットワークの両方から分離されたプライベート仮想ネットワークを作成できます。たとえば、アクセスがネットワーク全体にではなく、企業の開発者のみに制限されたネットワーク環境を作成できます。etherstubを使用すると、このような環境を作成できます。

etherstub と VNICは、Oracle Solaris の仮想化機能の一部にすぎません。通常、これらのコンポーネントは Oracle Solaris のコンテナまたはゾーンとともに使用します。
す。VNICまたはetherstubをゾーンで使用するように割り当てることによって、1つのシステム内にネットワークを作成できます。

▼仮想ネットワークインタフェースを作成する方法
この手順は、仮想ネットワークインタフェースカード(VNIC)を作成する方法を示しています。

1 管理者になります。
詳細は、『Oracle Solarisの管理:セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 (省略可能)システムの使用可能な物理インタフェースに関する情報を表示するには、次のコマンドを入力します。
```
# dladm show-phys
```
このコマンドは、システム上の物理NICとそれに対応するデータリンク名を表示します。データリンクのカスタマイズされた名前を作成しないかぎり、データリンクは、ネットワークインタフェースのデバイス名と同じ名前を持っています。たとえば、デバイスe1000g0は、リンク名が別の名前に置き換えられるまでデータリンク名e1000g0を使用します。カスタマイズされたデータリンク名についての詳細は、26ページの「ネットワークデバイスとデータリンク名」を参照してください。

3 (省略可能)システムのデータリンクに関する情報を表示するには、次のコマンドを入力します。
```
# dladm show-link
```
このコマンドは、各データリンクとその現在のステータスを一覧表示します。データリンクのSTATEフィールドに、そのデータリンクがupとして示されていることを確認してください。VNICは、ステータスがupであるデータリンク上のみ構成できます。

4 (省略可能)構成されているインタフェース上のIPアドレス情報を表示するには、次のコマンドを入力します。
```
# ipadm show-addr
```
このコマンドは、システム上に構成されているインタフェースとそれに対応するIPアドレスを一覧表示します。

5 データリンク上にVNICを作成します。
```
# dladm create-vnic -l link vnic
```
- linkは、そのVNICが構成されているデータリンクの名前です。
- vnicは、カスタマイズされた名前でもラベル付きできるVNICです。
6 リンク上にVNIC IPインタフェースを作成します。
  # ipadm create-ip vnic

7 VNIC に有効なIPアドレスを構成します。
静的IPアドレスを割り当てる場合は、次の構文を使用します。
  # ipadm create-addr -T static -a address addrobj
ここで、addrobj は、interface/user-defined-string という命名形式を使用します
(e1000g0/v4globalzなど)。このコマンドを使用するときのその他のオプションについて
は、ipadm(1M) のマニュアルページを参照してください。

8 静的IPアドレスを使用している場合は、/etc/hostsファイルにアドレス情報を追加
します。

9 (省略可能)VNIC のアドレス構成を表示するには、次のように入力します。
  # ipadm show-addr

10 (省略可能)VNIC の情報を表示するには、次のように入力します。
  # dladm show-vnic

例19-1 仮想ネットワークインタフェースの作成
この例では、VNICを作成するためのコマンドが含まれています。このコマンドを実
行するには、スーパーユーザーまたは同等の役割としてシステムにログインする必要
があります。

  # dladm show-phys
  LINK MEDIA STATE SPEED DUPLEX DEVICE
  net0 Ethernet up 1000 full e1000g0
  net1 Ethernet unknown 0 half e1000g1

  # dladm show-link
  LINK CLASS MTU STATE BRIDGE OVER
  net0 phys 1500 up -- --
  net1 phys 1500 unknown -- --

  # ipadm show-if
  IFNAME CLASS STATE ACTIVE OVER
  lo0 loopback ok yes --
  net0 ip ok yes --

  # ipadm show-addr
  ADDROBJ TYPE STATE ADDR
  lo0/7 static ok 127.0.0.1/8
  net0/v4addr static ok 192.168.3.70/24

  # dladm create-vnic -l net0 vnic0
  # dladm create-vnic -l net0 vnic1

  # dladm show-vnic
Oracle Solarisでのネットワーク仮想化のコンポーネントの構成

<table>
<thead>
<tr>
<th>LINK</th>
<th>OVER</th>
<th>SPEED</th>
<th>MACADDRESS</th>
<th>MACADDRTYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnic0</td>
<td>net0</td>
<td>1000 Mbps</td>
<td>2:8:20:c2:39:38</td>
<td>random</td>
</tr>
<tr>
<td>vnic1</td>
<td>net0</td>
<td>1000 Mbps</td>
<td>2:8:20:5f:84:ff</td>
<td>random</td>
</tr>
</tbody>
</table>

```
# ipadm create-ip vnic0
# ipadm create-ip vnic1

# ipadm create-addr -T static -a 192.168.3.80/24 vnic0/v4address
# ipadm create-addr -T static -a 192.168.3.85/24 vnic1/v4address

# ipadm show-addr
ADDROBJ TYPE STATE ADDR
lo0/? static ok 127.0.0.1/8
net0/v4addr static ok 192.168.3.70/24
vnic0/v4address static ok 192.168.3.80/24
vnic1/v4address static ok 192.168.3.85/24
```

システムの/etc/hostsファイルには、次のような情報が含まれます。

```
# cat /etc/hosts
#
::1 localhost
127.0.0.1 localhost
192.168.3.70 loghost #For e1000g0
192.168.3.80 vnic1
192.168.3.85 vnic2
```

▼etherstubを作成する方法

etherstubは、仮想ネットワークをシステム内のほかの仮想ネットワークや、システムが接続されている外部ネットワークから分離するために使用します。etherstubをそれ自体だけで使用することはできません。代わりに、VNICをetherstubとともに使用して、プライベート（つまり、分離された）仮想ネットワークを作成します。etherstubは、いつでも必要ならだけ作成できます。また、VNICも、各etherstub上にいくつでも必要ならだけ作成できます。

1 管理者になります。
詳細は、『Oracle Solaris 的管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 etherstubを作成します。
```
# dladm create-etherstub etherstub
```

3 etherstub上にVNICを作成します。
```
# dladm create-vnic -l etherstub vnic
```

4 VNICにプライベートアドレスを構成します。
注 - etherstub 上に VNIC を構成しているときの対象のネットワークを分離するには、外部ネットワークのデフォルトルーターで転送できないプライベート IP アドレスを使用するようにしてください。たとえば、物理インタフェースのアドレスが 192.168.3.0/24 であるとします。これは、システムが 192.168.3.x ネットワーク上には存在することを示します。そのため、デフォルトルーターには知られていない別のあるアドレス（たとえば、192.168.0.x）を割り当ててください。

5 (省略可能) VNIC に関する情報を表示するには、次のコマンドを入力します。

```bash
# dladm show-vnic
```

このコマンドは、システム内のすべての VNIC と、それらの VNIC の作成に使用されたデータリンクまたは etherstub を一覧表示します。

6 (省略可能) システム上のすべての物理および仮想リンクに関する情報を表示するには、次のコマンドを入力します。

```bash
# dladm show-link
```

例 19-2 etherstub の作成

次の例は、etherstub を作成したあと、その etherstub 上に VNIC を構成する方法を示しています。この例では、etherstub 上に構成される 3 番目の VNIC を追加することによって前の例を発展させています。

次のコマンドを実行するには、スーパーユーザーまたは同等の役割としてシステムにログインする必要があります。

```bash
# dladm create-etherstub stub0
# dladm show-vnic
LINK OVER SPEED MACADDRESS MACADRRTYPE
vnice1 net0 1000 Mbps 2:8:20:c2:39:38 random
vnice2 net0 1000 Mbps 2:8:20:5f:84:ff random
#
# dladm create-vnic -l stub0 vnic3
# ipadm create-vnic vnic3
# ipadm create-addr -T static -a 192.168.0.10/24 vnic3/privaddr
# dladm show-vnic
LINK OVER SPEED MACADDRESS MACADRRTYPE
vnice1 net0 1000 Mbps 2:8:20:c2:39:38 random
vnice2 net0 1000 Mbps 2:8:20:5f:84:ff random
vnice3 stub0 1000 Mbps 2:8:20:54:f4:74 random
#
# ipadm show-addr
ADDROBJ TYPE STATE ADDR
lo0/7 static ok 127.0.0.1/8
net0/v4addr static ok 192.168.3.70/24
vnic1/v4address static ok 192.168.3.80/24
vnic2/v4address static ok 192.168.3.85/24
vnic3/privaddr static ok 192.168.0.10/24
```

Oracle Solaris でのネットワーク仮想化のコンポーネントの構成
システムの/etc/hosts ファイルには、次のような情報が含まれます。

```
# cat /etc/hosts
#
::1 localhost
127.0.0.1 localhost
192.168.3.70 loghost  # For e1000g0
192.168.3.80 vnic1
192.168.3.85 vnic2
192.168.0.10 vnic3
```

### VNIC とゾーンの操作

この節では、ネットワーク仮想化コンポーネントをゾーンで使用されるように構成することによって、これらのコンポーネントを配備する方法を示します。この節では、VNICを使用するようにゾーンを操作する場合の次の2つのアプローチについて説明します。

- まったく新しいゾーンを作成し、これらのゾーン上にVNICを構成します。
- 既存のゾーン構成をVNICを使用するように変更します。

システムにはじめてログインすると、自動的にそのシステムの大域ゾーンに移動します。この大域ゾーン上にVNICを作成します。次に、これらのVNICが大域ゾーン、または排他的なタイプの非大域ゾーンのどちらかで使用されるかに応じて、これらのVNICをさらに構成します。ゾーンの概要については、『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』の「ゾーンの概要」を参照してください。

### VNIC とともに使用するための新しいゾーンの作成

このアプローチは、システム内に構成されているゾーンが存在しない場合、またはVNICを使用するための新しいゾーンを作成する場合に使用します。

VNICを使用するには、ゾーンを排他的IPゾーンとして構成する必要がありま
す。次の手順では、vnic1 とともに zone1 を構成します。zone2 を構成するため
に、同じ手順を実行する必要があります。わかりやすくするために、特定のコマンドがどのゾーンで発行されているかをプロンプトで示しています。ただし、プロンプトに表示される実際のパスは、特定のシステムのプロンプト設定に応じて異なることがあります。

▼ 排他的IPゾーンを作成および構成する方法

ゾーンを作成するときは、いくつかのパラメータを設定できます。この章全体にわたるゾーンの手順は、そのゾーンをVNICとともに動作できるようにすることに関連するパラメータのみに焦点を合わせています。ゾーン構成についての詳細は、『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』のパート II 「Oracle Solaris ゾーン」を参照してください。
始める前に  次の作業を完了していることを確認します。

- 371 ページの「仮想ネットワークインタフェースを作成する方法」の説明に従って、ゾーンのための VNIC を作成します。
- ゾーン名を定義します。
- ゾーンのホームディレクトリを決定します。
- 特定のゾーンに関連付ける特定の VNIC を決定します。
- VNIC の IP アドレスを決定します。
- ゾーンに対して指定するルーターアドレスなどの、その他のネットワーク情報を取得します。

1  管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2  作成するゾーンごとに、次の手順を実行します。

a. ゾーン構成ユーティリティーを起動し、ゾーンを作成します。
   ```
global# zonecfg -z zone
zonecfg:zone> create
```

b. パラメータ zonepath を定義することによって、ホームディレクトリを設定します。
   ```
zonecfg:zone> set zonepath=/home/export/zone
```

c. 自動ブートを有効にします。
   ```
zonecfg:zone> set autoboot=true
```

d. ゾーンを別の IP ゾーンになるように構成します。
   ```
zonecfg:zone> set ip-type=exclusive
```

e. ゾーンのインタフェースを、指定された VNIC になるように設定します。
   ```
zonecfg:zone> add net
zonecfg:zone:net> set physical=vnic
zonecfg:zone:net> end
zonecfg:zone>
```

f. 設定を確認してコミットしたあと、ゾーン構成ユーティリティーを終了します。
   ```
zonecfg:zone> verify
zonecfg:zone> commit
zonecfg:zone> exit
```
   ```
global#
```

g. (省略可能) ゾーンの情報が正しいことを確認するには、次のように入力します。
   ```
global# zonecfg -z zone info
```
注 - 次のように入力することによって、ゾーン構成ユーティリティーの実行中にも同じ情報を表示できます。

```
zonecfg:zone> info
```

3 ゾーンをインストールします。
```
global# zoneadm -z zone install
```

注 - インストール処理には時間がかかることがあります。

4 (省略可能) ゾーンが完全にインストールされ、そのゾーンのステータスを確認します。
```
zoneadm list -iv
```

注 - `-iv`オプションを指定すると、実行中かどうかは関係なく、構成されているすべてのゾーンが一覧表示されます。この段階では、今作成したゾーンのステータスは「running」ではなく、「installed」になります。`-v`オプションを使用すると、実行中のゾーンのみが一覧表示され、今作成したゾーンは除外されます。

5 ゾーンを起動します。
```
global# zoneadm -z zone boot
```

6 (省略可能) ゾーンが現在実行中であることを確認します。
```
zoneadm list -v
```

7 ゾーンが完全に起動したら、そのゾーンのコンソールに接続します。
```
# zlogin -c zone
```

8 入力を求められたら、情報を指定します。
いくつかの情報として、端末タイプ、領域、言語などがあります。ほとんどの情報は、選択リストから選択することによって指定されます。通常は、システム構成によって要求されないかぎり、デフォルトオプションで十分です。
指定または確認する必要のある、現在の手順に関連する情報は次のとおりです。
- ゾーンのホスト名(たとえば、zone1)。
- ゾーンのVNICのIPアドレスに基づいたゾーンのIPアドレス。
- IPv6を有効にするかどうか。
- 仮想ネットワークを含むシステムがサブネットの一部であるかどうか。
- IPアドレスのネットマスク。
例 19-3  ゾーンとVNICを作成することによる基本的な仮想ネットワークの構成

この例では、以前に仮想ネットワークを構成するためにゾーンとVNICを作成するときに示されたすべての手順が統合されています。この例では、サンプルのゾーンとしてzone1を使用しています。

この例は、次の前提に基づいています。

- **VNIC:** vnic1
- **ゾーン名:** zone1
- **ゾーンのホームディレクトリ:** /home/export/zone-name
- **VNICのゾーン割り当て:** zone1にvnic1
- **IPアドレス:** vnic1は192.168.3.80を使用
- **物理インタフェースのIPアドレス:** 192.168.3.70
- **ルーティアドレス:** 192.168.3.25

```
global# dladm show-phys
  LINK   MEDIA   STATE  SPEED  DUPLEX  DEVICE
  net0   Ethernet up   1000    full  e1000g0
  net1   Ethernet unknown 1000 full  bge0

global# dladm show-lnk
  LINK   CLASS  MTU  STATE  BRIDGE  OVER
  net0  phys   1500 up  --  --
  net1  phys   1500 unknown --  --

global# ipadm show-if
  IFNAME  CLASS  STATE  ACTIVE  OVER
  lo0    loopback ok    yes  --
  net0   ip       ok    yes  --

global # ipadm show-addr
  ADDROBJ  TYPE  STATE  ADDR
  lo0/7?  static  ok  127.0.0.1/8
  net0/v4addr static ok 192.168.3.70/24

global # dladm create-vnic -l net0 vnic1

global # dladm show-vnic
  LINK  OVER  SPEED  MACADDRESS  MACADORTYPE
  vnic1 net0 1000 Mbps 2:8:20:5f:84:ff random

global # ipadm create-ip vnic1

global # ipadm create-address -T static -a 192.168.3.80/24 vnic1/v4address

global # ipadm show-address
  ADDROBJ  TYPE  STATE  ADDR
  lo0/7?  static  ok  127.0.0.1/8
  net0/v4addr  static ok 192.168.3.70/24
```
VNICとゾーンの操作

vnic1/v4address  static  ok  192.168.3.80/24

global # cat /etc/hosts
::1 localhost
127.0.0.1 localhost
192.168.3.70 loghost  #For net0
192.168.3.80 zone1  #using vnic1

global # zonecfg -z zone1
zonecfg:zone1> create
zonecfg:zone1> set zonepath=/export/home/zone1
zonecfg:zone1> set autoboot=true
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic1
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> info
zonename: zone1
zonepath: /export/home/zone1
brand: native
autoboot: true
net:
    address not specified
    physical: vnic1
zonecfg:zone1> commit
zonecfg:zone1> exit

global#
global# zoneadm -z zone1 verify
WARNING: /export/home/zone1 does not exist, so it could not be verified.
When 'zoneadm install' is run, 'install' will try to create
/export/home/zone1, and 'verify' will be tried again,
but the 'verify' may fail if:
the parent directory of /export/home/zone1 is group- or other-writable
or
/export/home/zone1 overlaps with any other installed zones.

global# zoneadm -z zone1 install
Preparing to install zone <zone1>
Creating list of files to copy from the global zone.
.
.
Zone <zone1> is initialized.

global# zoneadm list -iv
<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td></td>
<td>zone1</td>
<td>installed</td>
<td>/export/home/zone1</td>
<td>native</td>
<td>excl</td>
</tr>
</tbody>
</table>

global# zoneadm -z zone1 boot
global# zoneadm list -v
<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>1</td>
<td>zone1</td>
<td>running</td>
<td>/export/home/zone1</td>
<td>native</td>
<td>excl</td>
</tr>
</tbody>
</table>

zlogin -C zone1
What type of terminal are you using?
VNICとゾーンの操作

8) Sun Workstation
9) Televideo 910
10) Televideo 925
11) Wyse Model 50
12) X Terminal Emulator (xterms)
13) CDE Terminal Emulator (dtterm)
14) Other

Type the number of your choice and press Return: 13

Hostname: zone1
IP address: 192.168.3.80
System part of a subnet: Yes
Netmask: 255.255.255.0
Enable IPv6: No
Default route: 192.168.3.70
Router IP address: 192.168.3.25

次の手順

さまざまなツールを使用してネットワークトラフィックを監視し、ゾーン使用状況に関する統計情報を取得することができます。

- ネットワークが正しく構成されていることを確認するには、『Oracle Solaris の管理: IP サービス』の第 5 章「TCP/IP ネットワークの管理」を参照してください。

- ネットワーク経由でトラフィックを監視するには、『Oracle Solaris の管理: IP サービス』の「snoop コマンドによるパケット転送の監視」を参照してください。

- ネットワークでのシステムリソースの使用方法を管理するには、第 21 章「ネットワークリソースの管理」を参照してください。

- アカウンティングの目的で統計情報を取得するには、第 22 章「ネットワークトラフィックとリソース使用状況の監視」を参照してください。

仮想ネットワークを分解する必要がある場合は、387 ページの「ゾーンを削除することなく仮想ネットワークを削除する方法」を参照してください。

既存のゾーンの構成を VNIC を使用するように変更する

このアプローチは、既存のゾーンで VNIC を使用するようにしたい場合に使用します。この場合、ゾーンにはすでにゾーン名が付けられ、それらのホームディレクトリまたは zonepaths がすでに定義されています。
ゾーンを VNIC を使用するように再構成する方法

始める前に
次の作業を完了していることを確認します。

- 371 ページの「仮想ネットワークインタフェースを作成する方法」の説明に従って、ゾーンのための VNIC を作成します。
- 特定のゾーンに関連付けられる特定の VNIC を決定します。
- VNIC の IP アドレスを決定します。
- ゾーンに対して指定するルーターアドレスなどの、その他のネットワーク情報を取得します。

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 ゾーンが正しく構成され、システム上で実行されていることを確認します。
```bash
global# zoneadm list -v
```
注：-v オプションを指定すると、実行中のゾーンのみが一覧表示されます。まだ起動されていないゾーンを含む、構成されているすべてのゾーンを一覧表示するには、-iv オプションを使用します。

3 VNIC とともに構成するゾーンごとに、次の手順を実行します。

a. ゾーンに関する情報を確認します。
```bash
global# zonecfg -z zone info
```
IP テイプとネットワークインタフェースに関する情報を確認します。ネットワークインタフェースは、パラメータ `physical` で指定されます。VNIC とともに構成されるゾーンの場合、そのゾーンは他ゾーンの IP ゾーンである必要があり、またネットワークインタフェースは VNIC を指定する必要があります。

b. 必要な場合は、共有ゾーンを排他的 IP ゾーンに変更します。
```bash
global# zonecfg -z zone
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1>
```

b. ゾーンのインタフェースを VNIC を使用するように変更します。
```bash
zonecfg:zone1> remove net physical=non-vnic-interface
zonecfg:zone1> add net
zonecfg:zone1> add net physical=vnic
zonecfg:zone1> set physical=vnic
zonecfg:zone1> end
```

d. 必要に応じて、その他のパラメータ値を変更します。

第19章 仮想ネットワークの構成(タスク) 381
e. 実装した変更を確認してコミットしたあと、そのゾーンを終了します。

zonecfg:zone1 verify
zonecfg:zone1> commit
zonecfg:zone1> exit
global#

f. ゾーンを再起動します。

global# zoneadm -z zone reboot

g. ゾーンがリプートしたあと、ip-typeとphysicalに関するゾーン情報が正しいことを見確認します。

global# zonecfg -z zone info ip-type
global# zonecfg -z zone info net
この情報に、そのゾーンのIPタイプが期待されるものであるなら、かつ指定されたVNICを使用していることが示されている必要があります。

4 ゾーンにログインします。

global# zlogin zone

5 VNICに有効なIPアドレスを構成します。
VNICに静的アドレスを割り当てる場合は、次を入力します。

zone# ipadm create-addr -T static -a address addrobj
ここで、addressがCIDR記号を使用できるのに対して、addrobjはinterface/user-defined-stringの命名規則に従います。

6 (省略可能)ゾーン内のインタフェース構成を確認します。
zone# ipadm show-if
または

zone# ipadm show-addr

例 19-4 ゾーン構成をVNICを使用するように変更することによる基本的な仮想ネットワークの構成

この例では、前の例と同じシステムを使用し、同じ前提条件に基づきます。このシステムでは、zone2がすでに共有ゾーンとして存在しているとします。zone2を使用するように変更しようとしています。

global# dladm show-link
<table>
<thead>
<tr>
<th>LINK</th>
<th>CLASS</th>
<th>MTU</th>
<th>STATE</th>
<th>BRIDGE</th>
<th>OVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>net0</td>
<td>phys</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>net1</td>
<td>phys</td>
<td>1500</td>
<td>unknown</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>vnic1</td>
<td>vnic</td>
<td>1500</td>
<td>up</td>
<td>--</td>
<td>e1000g0</td>
</tr>
</tbody>
</table>

global# ipadm show-if
IFNAME  CLASS  STATE  ACTIVE  OVER
lo0    loopback  ok  yes  --
net0   ip  ok  yes  --
vnic1  ip  ok  yes  --

global # ipadm show-addr
ADDROBJ  TYPE  STATE  ADDR
lo0/7  static  ok  127.0.0.1/8
net0/v4addr  static  ok  192.168.3.70/24
vnic1/v4address  static  ok  192.168.3.80/24

global # dladm create-vnic -l net0 vnic2
global # dladm show-vnic
LINK  OVER  SPEED  MACADDRESS  MACADDRTYPE
vnic1  net0  1000 Mbps  2:8:20:5f:84:ff  random
vnic2  net0  1000 Mbps  2:8:20:54:f4:74  random

global# zoneadm list -v
ID  NAME  STATUS  PATH  BRAND  IP
0  global  running  /  native  shared
1  zone1  running  /export/home/zone1  native  excl
2  zone2  running  /export/home/zone2  native  shared

global# zonecfg -z zone2 info
zonename: zone2
zonepath: /export/home/zone2
brand: native
autoboot: true
bootargs:
pool: z2-pool
limitpriv:
scheduling-class:
ip-type: shared
hostid:
inherit-pkg-dir:
dir: /etc/crypto
net:
  address not specified
physical: e1000g0
defrouter not specified

global#
global# zonecfg -z zone2
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1> remove net physical=net0
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic2
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
VNICとゾーンの操作

```
zonecfg:zone1> exit
global#

global# zonecfg -z zone2 info ip-type
ip-type: exclusive
global#

global# zonecfg -z zone2 info net
net:
    address ot specified
    physical: vnic2
    defrouter not specified
global#

global# zlogin zone2
zone2# ipadm create-ip vnic2
zone2# ipadm create-addr -T static -a 192.168.3.85/24 vnic2/v4address

zone2# ipadm show-addr
  ADDR OBJ TYPE STATE ADDR
  lo0/v4 static ok 127.0.0.1/8
  vnic2/v4address static ok 192.168.3.85/24

zone1# exit
global#

global# vi /etc/hosts
#
::1 localhost
127.0.0.1 localhost
192.168.3.70 loghost #For e1000g0
192.168.3.80 zone1 #using vnic1
192.168.3.85 zone2 #using vnic2
```

次の手順：ネットワークの設定をさらに構成してシステムリソースの使用をカスタマイズするか、またはさまざまなツールを使用してネットワークトラフィックを監視し、リソース使用状況に関する統計情報を取得することができます。

- ネットワークが正しく構成されていることを確認するには、次を参照してください。
- ネットワーク経由でトラフィックを監視するには、次を参照してください。
- ネットワークでのシステムリソースの使用方法を管理するには、次を参照してください。
- アカウントの目的で統計情報を取得するには、次を参照してください。

仮想ネットワークを分解する必要がある場合は、387ページの「ゾーンを削除することなく仮想ネットワークを削除する方法」を参照してください。

Oracle Solaris管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
プライベート仮想ネットワークの作成

この節の例では、1つのシステム上でプライベート仮想ネットワークを構成する方法を示します。プライベート仮想ネットワークは、仮想プライベートネットワーク (VPN) とは異なります。VPN ソフトウェアは、2つのエンドポイントシステム間のセキュリティ保護されたポイントツーポイントリンクを作成します。この節のタスクで構成されるプライベートネットワークは、外部システムからはアクセスできないボックス上の仮想ネットワークです。

プライベートネットワークのゾーンがホストを超えてパケットを送信できるようにするために、ネットワークアドレス変換 (NAT) デバイスを構成します。NAT は、VNIC のプライベート IP アドレスを物理ネットワークインタフェースのルーティング可能な IP アドレスに変換しますが、プライベート IP アドレスが外部ネットワークに公開されることはありません。また、次の例にはルーティング構成も含まれています。

例19-5 プライベート仮想ネットワーク構成の作成

次の例では、前の例と同じシステムを使用し、同じ前提条件下で進めます。具体的には、zone1 と zone2 は現在、仮想ネットワークとして構成されています。システム内に zone3 がすでに存在しています。zone3 を、ネットワークのほかの部分から分離されたプライベートネットワークになるように変更します。NAT と IP 転送を構成して仮想プライベートネットワークがホストの外部にパケットを送信できるようにしますが、引き続き外部ネットワークからはプライベートアドレスが見えないようにします。

global# dladm create-etherstub stub0

global# dladm create-vnic -l etherstub0 vnic3

global# dladm show-vnic

<table>
<thead>
<tr>
<th>LINK</th>
<th>OVER</th>
<th>SPEED</th>
<th>MACADDRESS</th>
<th>MACADDRTYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnic1</td>
<td>net0</td>
<td>1000 Mbps</td>
<td>2:8:20:5f:84:ff</td>
<td>random</td>
</tr>
<tr>
<td>vnic2</td>
<td>net0</td>
<td>1000 Mbps</td>
<td>2:8:20:54:f4:74</td>
<td>random</td>
</tr>
<tr>
<td>vnic3</td>
<td>stub0</td>
<td>0 Mbps</td>
<td>2:8:20:6b:8:ab</td>
<td>random</td>
</tr>
</tbody>
</table>

global# vi /etc/hosts

```
#
::1 localhost
127.0.0.1 localhost
192.168.3.70 loghost #For e1000g0
192.168.3.80 zone1 #using vnic1
192.168.3.85 zone2 #using vnic2
```

この段階で、zone3 を vnic3 上の他者の IP ソーンになるように変更します。

global# zonecfg -z zone3
zonecfg:zone3> set ip-type=exclusive
zonecfg:zone3> remove net physical=e1000g0
zonecfg:zone3> add net
zonecfg:zone3:net> set physical=vnic3
zonecfg:zone3:net> end
例19-5 プライベート仮想ネットワーク構成の作成 (続き)

```
zonecfg:zone3> vereify
zonecfg:zone3> commit
zonecfg:zone3> exit

global#

zonestart zone3

ipadm create-ip vnic3
ipadm create-addr -T static -a 192.168.0.10/24 vnic3/privaddr

zone3# ipadm show-addr
ADDROBJ TYPE STATE ADDR
lo0/v4 static ok 127.0.0.1/8
vnic3/privaddr static ok 192.168.0.10/24
zone3# exit
```

```
vi /etc/hosts
::1 localhost
127.0.0.1 localhost
192.168.3.70 loghost #For e1000g0
192.168.3.80 zone1 #using vnic1
192.168.3.85 zone2 #using vnic2
192.168.0.10 zone3 #using vnic3
```

```
routeadm
Configuration Current Current
Option Configuration System State
-------------------------------
IPv4 routing enabled enabled
IPv6 routing disabled disabled
IPv4 forwarding disabled disabled
IPv6 forwarding disabled disabled

Routing services "route:default ripng:default"
```

```
global# ipadm set-ifprop -p forwarding=yes -m ipv4 e1000g0
```

```
vi /etc/ipf/ipnat.conf
```

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
例19-5 プライベート仮想ネットワーク構成の作成 （続き）

map e1000g0 192.168.0.0/24 -> 0/32 portmap tcp/udp auto
map e1000g0 192.168.0.0/24 -> 0/32
global# svcadm enable network/ipfilter
global# zoneadm -z zone1 boot
global# zoneadm -z zone2 boot
global# zoneadm -z zone3 boot

▼ ゾーンを削除することなく仮想ネットワークを削除する方法

次の手順は、ゾーンの仮想ネットワークを無効にするが、そのゾーンはそのままの状態にしておくための方法を示しています。

この手順は、次のいずれかを行う必要がある場合に使用します。

- 既存のゾーンを別の構成で使用します。たとえば、あるゾーンを、etherstubを使用してゾーンを作成する必要のあるプライベートネットワークの一部として構成することが必要になる場合があります。
- ゾーンを別のネットワークに移行します。
- ゾーンを別のゾーンパスに移動します。
- ゾーンを複製します。この操作は、『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』の「同一システム上での非大域ゾーンのクラーニング」の説明に従って行います。

始める前に このタスクでは、排他的 IP ゾーンで構成された実行中の仮想ネットワークが存在することを前提にしています。

1 管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 現在構成されているゾーンの状態を確認します。

  # zoneadm list -v

次のような情報が表示されます。

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>1</td>
<td>zone1</td>
<td>running</td>
<td>/export/home/zone1</td>
<td>native</td>
<td>excl</td>
</tr>
<tr>
<td>2</td>
<td>zone2</td>
<td>running</td>
<td>/export/home/zone2</td>
<td>native</td>
<td>excl</td>
</tr>
<tr>
<td>3</td>
<td>zone3</td>
<td>running</td>
<td>/export/home/zone3</td>
<td>native</td>
<td>excl</td>
</tr>
</tbody>
</table>
3 仮想ネットワークの排他のIPゾーンを停止します。
停止するゾーンごとに、次のコマンドを別々に発行します。

```
global# zoneadm -z zone-name halt
```

ゾーンを停止したら、『Oracle Solarisのシステム管理(Oracle Solarisゾーン、Oracle Solaris 10ゾーン、およびリソース管理)』の「ゾーンを停止する」の説明に従って、そのゾーンのアプリケーション環境を削除し、いくつかのシステム動作を終了します。

4 ゾーンが停止されていることを確認します。

```
# zoneadm list -iv
```

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>zone1</td>
<td>installed</td>
<td>/export/home/zone1</td>
<td>native</td>
<td>excl</td>
</tr>
<tr>
<td>-</td>
<td>zone2</td>
<td>installed</td>
<td>/export/home/zone2</td>
<td>native</td>
<td>excl</td>
</tr>
<tr>
<td>-</td>
<td>zone3</td>
<td>installed</td>
<td>/export/home/zone3</td>
<td>native</td>
<td>excl</td>
</tr>
</tbody>
</table>

ゾーンは実行されなくなりましたが、インストールされたままであることに注意してください。停止されたゾーンをリブートするには、『Oracle Solarisのシステム管理(Oracle Solarisゾーン、Oracle Solaris 10ゾーン、およびリソース管理)』の「ゾーンのブート方法」を参照してください。

5 停止されたゾーン用に構成されていたVNICを一覧表示します。

```
# dladm show-vnic
```

<table>
<thead>
<tr>
<th>LINK</th>
<th>OVER</th>
<th>SPEED</th>
<th>MACADDRESS</th>
<th>MACADDRTYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnic1</td>
<td>net0</td>
<td>1000 Mbps</td>
<td>2:8:20:5f:84:ff</td>
<td>random</td>
</tr>
<tr>
<td>vnic2</td>
<td>net1</td>
<td>1000 Mbps</td>
<td>2:8:20:54:f4:74</td>
<td>random</td>
</tr>
<tr>
<td>vnic3</td>
<td>stub0</td>
<td>1000 Mbps</td>
<td>2:8:20:2c:39:38</td>
<td>random</td>
</tr>
</tbody>
</table>

結果の出力は、これらのVNICが引き続き、大域ゾーン内でデータリンクとして構成されていることを示します。ただし、それに対応するIPインタフェースは大域ゾーンではなく、これらのVNICが関連付けられているゾーン上で作成され、有効にされました。これらの非大域ゾーンは現在停止されています。

6 VNICを削除します。

```
# dladm delete-vnic vnic
```

たとえば、図18-1にあるゾーン内のVNICを削除するには、次を入力します。

```
# dladm delete-vnic vnic1
# dladm delete-vnic vnic2
```
仮想化環境でのリンク保護の使用

この章では、リンク保護について、およびOracle Solaris システム上でリンク保護を構成する方法について説明します。この章の内容は次のとおりです。

- 389 ページの「リンク保護の概要」
- 391 ページの「リンク保護の構成（タスクマップ）」

リンク保護の概要

システム構成で仮想化がますます多く採用されるにつれて、ゲスト仮想マシン (VM) に、ホスト管理者が物理または仮想リンクへの排他的アクセスを許可する場合があります。この構成により、仮想環境のネットワークトラフィックを、ホストシステムによって送受信されるより広範囲のトラフィックから切り離すことが可能になるため、ネットワークパフォーマンスが向上します。同時に、この構成によって、ゲスト環境によって生成される可能性のある有害なパケットのリスクにシステムやネットワーク全体がさらされるおそれがあります。

リンク保護は、潜在的に悪意のあるゲスト VM のためにネットワークで発生する可能性のある損害を回避することを目的にしています。この機能は、次の基本的な脅威からの保護を実現します。

- IP と MAC のなりすまし
- BPDU (ブリッジプロトコルデータユニークト) 攻撃などの、L2 フレームのなりすまし

注 - 特に、フィルタリングの要件が複雑な構成の場合、リンク保護でファイアウォールの配備を置き換えるべきではありません。
リンク保護タイプ

リンク保護メカニズムは、デフォルトでは無効になっています。リンク保護を有効にするには、protectionリンクプロバティーの値として次の保護タイプの1つまたは複数を指定します。

mac-nospoof MACのなりすまからの保護を有効にします。送信パケットの発信元MACアドレスが、そのデータリンクの構成されたMACアドレスに一致している必要があります。それ以外の場合、パケットは破棄されます。そのリンクがゾーンに属している場合は、mac-nospoofを有効にすると、ゾーンの所有者がそのリンクのMACアドレスを変更できなくなります。

ip-nospoof IPのなりすまからの保護を有効にします。IP、ARP、またはNDPの送信パケットにはすべて、DHCPで構成されたIPアドレスか、またはallowed-ipsリンクプロバティーに示されているいずれかのアドレスのどちらかに一致するアドレスフィールドが含まれている必要があります。それ以外の場合、パケットは破棄されます。

allowed-ips リンクプロバティーは、ip-nospoof保護タイプと連携します。デフォルトでは、このプロバティーによって指定される一覧は空です。このプロバティーが空か、または構成されていない場合は、次のIPアドレスが暗黙にこのプロバティーに含まれます。これらのIPアドレスは送信パケットのIPアドレスと照合され、そのパケットが通過を許可されるか、または破棄されるかが判定されます。

- 動的に学習されたDHCPで構成されたIPv4またはIPv6アドレス
- RFC2464に準拠し、リンクのMACアドレスから派生するリンクローカルIPv6アドレス

次の一覧は、プロトコルと、allowed-ipsプロバティー内のアドレスに一致する必要のある対応する送信パケットの関連付けられたアドレスフィールドを示しています。このプロバティーが空の場合、パケットのアドレスは、DHCPで構成されたIPアドレスに一致する必要があります。

- IP(IPv4またはIPv6)-パケットの発信元アドレス
- ARP-パケットの送信側プロトコルアドレス

restricted 送信パケットを、IPv4、IPv6、およびARPプロトコルタイプのパケットのみに制限します。ここに示されているタイプではないその他のパケットは破棄されます。この保護タイプを使用すると、リンクが、有害なおそれのあるL2制御フレームを生成できなくなります。
リンク保護の構成 (タスクマップ)

リンク保護を使用するには、dladm コマンドのいずれかのオプションを使用してリンクプロパティーを設定します。保護のタイプがほかの構成ファイルと連携する場合は、たとえば (ip-nospoof および allowed-ips)、次の 2 つの一般的なアクションを実行します。最初に、リンク保護を有効にします。次に、その構成ファイルをカスタマイズして、リンク保護の動作を決定します。

次の表は、Oracle Solaris サーバー上でリンク保護を構成するために使用できる各タスクを示しています。

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>リンク保護メカニズムを有効にします。</td>
<td>dladm set-linkprop コマンドを使用して、あるリンクのリンク保護タイプを有効にします。</td>
<td>392 ページの「リンク保護メカニズムを有効にする方法」</td>
</tr>
<tr>
<td>リンク保護メカニズムを無効にします。</td>
<td>dladm reset-linkprop コマンドを使用して、リンク保護を無効にします。</td>
<td>392 ページの「リンク保護を無効にする方法」</td>
</tr>
<tr>
<td>IP リンク保護タイプをカスタマイズします。</td>
<td>dladm set-linkprop コマンドを使用して、allowed-ips プロパティーの値を構成または変更します。</td>
<td>392 ページの「IP のなりすましからの保護のための IP アドレスを指定する方法」</td>
</tr>
<tr>
<td>リンク保護構成を表示します。</td>
<td>dladm show-linkprop コマンドを使用して、protection および allowed-ips プロパティー名を指定することによりリンク保護構成を表示します。</td>
<td>393 ページの「リンク保護構成を表示する方法」</td>
</tr>
</tbody>
</table>
リンク保護の構成(タスクマップ)

▼ リンク保護メカニズムを有効にする方法
この手順では、mac-nospoof、ip-nospoof、およびrestrictedのリンク保護タイプの1つまたは複数を有効にします。

1 管理者になります。
詳細は、「Oracle Solarisの管理: セキュリティサービス」の「管理権限を取得する方法」を参照してください。

2 1つ以上の保護タイプを指定することによってリンク保護を有効にします。
# dladm set-linkprop -p protection=value[,value,...] link
次の例では、vnic0リンク上で3つのすべてのリンク保護タイプを有効にします。
# dladm set-linkprop -p protection=mac-nospoof,ip-nospoof,restricted vnic0

▼ リンク保護を無効にする方法
この手順では、リンク保護をデフォルト値にリセットします。これにより、リンク保護が無効になります。

1 管理者になります。
詳細は、「Oracle Solarisの管理: セキュリティサービス」の「管理権限を取得する方法」を参照してください。

2 protectionプロパティをデフォルト値にリセットすることによってリンク保護を無効にします。
# dladm reset-linkprop -p protection link

▼ IPのなりすましからの保護のためのIPアドレスを指定する方法
allowed-ipsプロパティは、protectionプロパティでip-nospoof保護タイプが有効になっている場合にのみ使用されることに注意してください。

1 管理者になります。
詳細は、「Oracle Solarisの管理: セキュリティサービス」の「管理権限を取得する方法」を参照してください。
リンク保護の構成(タスクマップ)

2 IPのなりすましからの保護を有効にしていることを確認します。
このタイプのリンク保護をまだ有効にしていない場合は、次のコマンドを発行します。

```bash
# dladm set-linkprop -p protection=ip-nospoof
```

3 `allowed-ips` リンクプロバティーの値としてIPアドレスのリストを指定します。

```bash
# dladm set-linkprop -p allowed-ips=IP-addr1,IP-addr2,... link
```
次の例は、vnic0リンクに対するallowed-ipsプロバティーの値として`10.0.0.1`と`10.0.0.2`のIPアドレスを指定する方法を示しています。

```bash
# dladm set-linkprop -p allowed-ips=10.0.0.1,10.0.0.2 vnic0
```

▼ リンク保護構成を表示する方法

`protection`および`allowed-ips`プロバティーの値は、リンク保護の構成方法を示しています。`allowed-ips`プロバティーは、`protection`プロバティーで`ip-nospoof`保護タイプが指定されている場合にのみ使用されることに注意してください。

1 管理者になります。
詳細は、『Oracle Solarisの管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 リンク保護のプロバティー値を表示します。

```bash
# dladm show-linkprop -p protection,allowed-ips link
```
次の例は、vnic0リンクに対するprotectionおよびallowed-ipsプロバティーの値を示しています。

```
# dladm show-linkprop -p protection,allowed-ips vnic0

<table>
<thead>
<tr>
<th>LINK</th>
<th>PROPERTY</th>
<th>PERM</th>
<th>VALUE</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnic0</td>
<td>protection</td>
<td>rw</td>
<td>ip-nospoof</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>mac-nospoof</td>
<td></td>
<td>restricted</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>vnic0</td>
<td>allowed-ips</td>
<td>rw</td>
<td>10.0.0.1,10.0.0.2</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
```
第21章
ネットワークリソースの管理

この章では、データリンク (VNICなどの仮想リンクを含む)上のリソースを管理する方法について説明します。ネットワークリソースの管理では、特に仮想ネットワークでのパフォーマンスを向上させるために、サービスの品質を実装します。

この章の内容は次のとおりです。
- 395ページの「ネットワークリソースの管理の概要」
- 398ページの「ネットワークリソースの管理(タスクマップ)」
- 399ページの「データリンク上のリソースの管理」
- 419ページの「フロー上のリソースの管理」

ネットワークリソースの管理の概要

この節では、ネットワークケーンを導入することによりネットワークリソースの管理について説明します。また、データリンクプロパティを設定することによってネットワークリソースの管理を実装する方法についても説明します。さらに、ネットワークトラフィックを処理するためにリソース制御をさらに設定する別の方法としてのフローも定義されています。

リソース制御のためのデータリンクプロパティ

以前のOracle Solarisリリースでは、サービスの品質の実装は複雑なプロセスです。このプロセスでは、キューニング規則、クラス、および振り分け規則を定義し、これらのすべてのコンポーネント間の関係を示します。詳細は、「Oracle Solarisの管理: IPサービス」のパートV「IPサービス品質(IPQoS)」を参照してください。

このリリースでは、ネットワークリソースを管理することによって、サービスの品質がより容易に、かつ動的に得られます。ネットワークリソースの管理では、ネットワークリソースに関連するデータリンクプロパティを設定します。これらのプロパティを設定することによって、特定のリソースのどれだけの量を
ネットワークリソースの管理の概要

ネットワークプロセスに使用できるかを判断します。たとえば、リンクを、ネットワークプロセスのために排他的に予約された特定の数のCPUに関連付けることができます。または、リンクに、特定のタイプのネットワークトラフィックを処理するための特定の帯域幅を割り当てることができます。リソースのプロパティが定義されると、ただちに新しい設定が有効になります。この方法によって、リソースの管理が柔軟になります。リソースのプロパティは、リンクを作成するときに設定できます。あるいは、たとえば一定期間にわたってリソース使用状況を調査して、リソースをより適切に割り当てる方法を特定したあとは、あとからこれらのプロパティを設定することもできます。リソースを割り当てるための手順は、仮想ネットワーク環境と従来の物理ネットワークの両方に適用されます。

ネットワークリソースの管理は、トラフィックのための専用のレーンを作成することとに相当します。特定のタイプのネットワークパケットの要求に基づいたためにさまざまなリソースを結合すると、これらのリソースによって、このパケットのためのネットワークレーンが形成されます。ネットワークレーンごとに、リソースを異なる方法で割り当てることができます。たとえば、ネットワークトラフィックの負荷がもっとも高いレーンに、より多くのリソースを割り当てることができます。ネットワークレーンを実際のニーズに応じて分散されるように構成することによって、パケットを処理するシステムの効率が向上します。ネットワークレーンについての詳細は、425ページの「ネットワークトラフィックフローの概要」を参照してください。

ネットワークリソースの管理は、次のタスクに役立ちます。

- ネットワークのプロビジョニング。
- サービスレベル契約の確立。
- クライアントへの請求。
- セキュリティーに関する問題の診断。

以前のリソースでの複雑なQoS規則の定義を必要とすることなく、個々のシステム上のデータトラフィックの分離、優先順位付け、追跡、制御などを行うことができます。

フローを使用することによるネットワークリソースの管理

フローは、パケットを分類することにより、これらのパケットの処理でのリソースの使用方法をさらに制御するためのカスタマイズされた方法です。ネットワークパケットは、属性に従って分類できます。属性を共有するパケットによってフローが構成され、これらのパケットには特定のフロー名が付かれます。次に、このフローに特定のリソースを割り当てることができます。

フローを作成するための基礎として機能する属性は、パケットのヘッダー内の情報から派生します。パケットトラフィックは、次の場合的情况下に従ってフローに構成できます。
ネットワークリソースの管理のためのコマンド

ネットワークリソースを割り当てるためのコマンドは、直接データリンクを操作しているのか、またはフローを操作しているのかによって異なります。

- データリンクの場合は、リンクの作成中にプロパティを設定しているのか、または既存のリンクのプロパティを設定しているのかに応じて、該当するdladmサブコマンドを使用します。リンクの作成と、そのリンクへのリソースの割り当てを同時に行うには、次の構文を使用します。

  # dladm create-vnic -l link -p property=value[,property=value] vnic

ここで、linkには物理リンクまたは仮想リンクを指定できます。

既存のリンクのプロパティを設定するには、次の構文を使用します。

  # dladm set-linkprop -p property=value[,property=value] link

dladmコマンドと、このコマンドによって管理されるプロパティについての詳細は、dladm(1M)のマニュアルページを参照してください。

リソース割り当てのために設定できるリンクプロパティを次に示します。

- 帯域幅 - 特定のリンク使用のためのハードウェアの帯域幅を制限できます。
- NICリング - NICでリング割り当てがサポートされている場合は、その送信リンクと受信リンクをデータリンクで専用に使用するために割り当てることができます。NICリングについては、399ページの「送信リングと受信リング」で説明されています。
ネットワークリソースの管理(タスクマップ)

- CPU プール - CPU のプールは一般に、作成されて特定のゾーンに関連付けられます。これらのプールをデータリンクに割り当てることにより、関連付けられたゾーンのネットワークプロセスを管理するために一連の CPU を予約できます。CPU とプールについては、414 ページの「プールと CPU」で説明されています。

- CPU - 複数の CPU を備えたシステムでは、指定された数の CPU を特定のネットワーク処理専用に割り当てることができます。

- フローの場合は、flowadm サブコマンドを使用します。最初に、flowadm add-flow サブコマンドを使用してフローを作成します。次に、flowadm set-flowprop サブコマンドを使用して、そのフローにリソースを割り当てます。フローを特徴づける一連の定義された属性がまとまって、システムのフロー制御ポリシーを構成します。

注 - フローに割り当てることのできるリソース割り当てのためのプロパティーは、リンクに直接割り当てられるプロパティーと同じです。ただし、現在、フローに関連付けることができるのは帯域幅プロパティーだけです。プロパティーを設定するためのコマンドはデータリンクの場合とフローの場合で異なりますが、構文は似ています。帯域幅プロパティーを構成するには、420 ページの「フローを構成する方法」にある例を参照してください。

詳細は、flowadm(1M) のマニュアルページを参照してください。

ネットワークリソースの管理 (タスクマップ)

次の表は、リソース制御を確立し、これらのリソースがネットワーク処理のために割り当てられる方法を決定するための各種の方法を示しています。

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC クライアントにリングを割り当てます。</td>
<td>データリンク上の MAC クライアントを、リングを使用するよう構成します。</td>
<td>400 ページの「リング割り当てのためのプロパティー」</td>
</tr>
<tr>
<td>データリンクに CPU のプールを割り当てます。</td>
<td>ゾーンのネットワークプロセスを管理するために一連の CPUを割り当てには、poolプロパティーを使用します。</td>
<td>416 ページの「CPU プールをデータリンク用に構成する方法」</td>
</tr>
<tr>
<td>定義されたデータリンクに一連の CPU を割り当てます。</td>
<td>複数の CPU を備えたシステム上で、ネットワーキングの目的に一連の CPU を予約します。</td>
<td>418 ページの「リンクに CPU を割り当てる方法」</td>
</tr>
</tbody>
</table>
データリンク上のリソースの管理

この節では、物理ネットワークまたは仮想ネットワークのどちらかのネットワークパフォーマンスを向上させるために設定できる、選択されたリンクプロパティについて説明します。

送信リングと受信リング

NIC上にある受信 (Rx) リングと送信 (Tx) リングはそれぞれ、システムがネットワークパケットの受信と送信を行うために使用するハードウェアリソースです。以下節では、リングの概要について説明したあと、ネットワークプロセスをリンクを割り当てるために使用される手順を示します。また、リングを割り当てるためのコマンドを発行したときのメカニズム動作を示す例も提供されます。

MACクライアントとリング割り当て

VNICやその他のデータリンクなどのMACクライアントは、システムとほかのネットワークノードの間の通信を可能にするためにNIC上に構成されます。各クライアントは構成されたあと、RxリングとTxリングの両方を使用して、それぞれネットワークパケットを受信または送信します。MACクライアントは、ハードウェアベースまたはソフトウェアベースのどちらかです。ハードウェアベースのクライアントは、次の条件のいずれかを満たします。

- 1つ以上のRxリングを専用に使用できます。
- 1つ以上のRxリングを専用に使用できます。
- 1つ以上のRxリングと1つ以上のTxリングを専用に使用できます。

これらの条件のいずれも満たさないクライアントは、ソフトウェアベースのMACクライアントと呼ばれます。

NICによっては、ハードウェアベースのクライアントにリングを排他的使用のため割り当てることができるです。nxgeなどのNICでは、動的なリング割り当てをサポートしています。このようなNICでは、ハードウェアベースのクライアントを構成できるだけではありません。リングが割り当てに使用できる状態のままにあると仮定すると、このようなクライアントに割り当てるリングの数を決定できる柔軟性
データリンク上のリソースの管理

もあります。リングの使用は常に、主インタフェース (たとえば、nxge0) のために最適化されます。主インタフェースは主クライアントとも呼ばれています。ほかのクライアントによって排他的使用のために割り当てられていな使用可能なリングはすべて、主インタフェースに自動的に割り当てられます。

ixge などのほかの NIC は、静的なリング割り当てのみをサポートしています。これの NIC では、ハードウェアベースのクライアントだけを作成できます。各クライアントには、クライアントごとに固定されたリングのセットが自動的に構成されます。この固定されたセットは、NIC ドライバの初期構成中に決定されます。静的なリング割り当てのためのドライバの初期構成についての詳細は、『Oracle Solaris カーネルのチューンアップ・リファレンスマニュアル』を参照してください。

### VLAN でのリング割り当て

VLAN でのリングの割り当て処理は、その VLAN が作成された方法によって異なります。VLAN は、次の 2 つの方法のどちらかで作成されます。

- `dladm create-vlan` サブコマンドを使用して:

  ```
  # dladm create-vlan -l link -v VID vlan
  ```

- `dladm create-vnic` サブコマンドを使用して:

  ```
  # dladm create-vnic -l link -v VID vnic
  ```

dladm create-vlan サブコマンドで作成された VLAN は、ベースとなるインタフェースと同じ MAC アドレスを共有します。その結果、その VLAN は、ベースとなるインタフェースの Rx リングと Tx リングも共有します。dladm create-vnic コマンドを使用して VNIC として作成された VLAN には、ベースとなるインタフェースとは異なる MAC アドレスが割り当てられます。このような VLAN へのリングの割り当ては、ベースとなるリンクへの割り当てとは独立して行われます。そのため、NIC がハードウェアベースのクライアントをサポートしていると仮定すると、その VLAN には独自の専用リングを割り当てることができます。

### リング割り当てのためのプロパティ

リングを管理するために、dladm コマンドを使用して、次の 2 つのリングプロパティを設定できます。

- `rxrings` は、指定されたリンクに割り当てられた Rx リングの数を示します。
- `txrings` は、指定されたリンクに割り当てられた Tx リングの数を示します。

各プロパティは次の 3 つの指定可能な値のいずれかに設定できます。

- `sw` は、ソフトウェアベースのクライアントを構成していることを示します。このクライアントは、リングを排他的に使用できません。代わりに、このクライアントは、同様に構成されたほかのすべての既存のクライアントとリングを共有します。
データリンク上のリソースの管理

- $n > 0$（0より大きい数）は、ハードウェアベースのクライアントの構成にのみ適用されます。この数は、クライアントに排他的使用のために割り当てるリングの数を示します。数を指定できるのは、ベースとなるNICが動的なリング割り当てをサポートしている場合だけです。

- hwもまた、ハードウェアベースのクライアントの構成に適用されます。ただし、このようなクライアントでは、専用リングの実際の数を指定することはできません。代わりに、NICドライバの初期構成に従って、クライアントごとに固定された数のリングがすでに設定されています。ベースとなるNICが静的なリング割り当てのみをサポートしている場合は、*ringsプロパティをhwに設定します。

現在のリングの割り当てと使用に関する情報を表示するために、次の追加の読み取り専用のリングプロパティが使用できます。

- rxrings-availableとtxrings-availableは、割り当てに使用できるRxリングとTxリングの数を示します。

- rxhwclnt-availableとtxhwclnt-availableは、NIC上に構成できるRxとTxのハードウェアベースのクライアントの数を示します。

ハードウェアベースのクライアントを構成するための準備
ハードウェアベースのクライアントを構成する前に、システム上のNICのリング割り当て機能について知っている必要があります。これらの必要な情報を取得するには、次のコマンドを使用します。

```
# dladm show-linkprop link
```
ここでの、linkは特定のNICのデータリンクを示します。

特定のプロパティを表示するには、次のコマンドを使用します。

```
# dladm show-linkprop -p property1,property,...] link
```
ハードウェアベースのクライアントを正しく構成するには、次のことを行う必要があります。

- NICがハードウェアベースのクライアントをサポートしているかどうか
  コマンド出力のrxringsおよびtxringsプロパティは、NICがハードウェアベースのクライアントをサポートしているかどうかを示します。また、同じデータから、NICでサポートされているリング割り当てのタイプを特定することもできます。

- ハードウェアベースのクライアントに割り当てるためのリングの可用性
  コマンド出力のrxrings-availableおよびtxrings-availableプロパティは、ハードウェアベースのクライアントに割り当てることができる使用可能なRxリングとTxリングを示します。
データリンク上のリソースの管理

- リンク上に構成できるハードウェアベースのクライアントの可用性

リングはセットとして割り当てられます。使用可能なリングの数と、専用リングを使用できるクライアントの数の間に一対一の対応関係は存在しません。その結果、リングを割り当てると、リングの可用性だけでなく、専用リングを使用するように引き続き構成できる追加のハードウェアベースのクライアントの数も確認する必要があります。リングを割り当てることができるのは、リングとハードウェアベースのクライアントの両方が使用可能な場合だけです。

コマンド出力の `rxhwclnt-available` および `txhwclnt-available` プロパティーは、専用の Rx リングと Tx リングを使用できる、構成可能なハードウェアベースのクライアントの数を示します。

NIC がリング割り当てをサポートしており、さらにリングとハードウェアベースのクライアントが使用可能な場合は、404 ページの「ハードウェアベースのクライアントを構成する方法」の説明に従って、システムにこのタイプのクライアントを構成できます。あるいは、代わりに 405 ページの「ソフトウェアベースのクライアントを作成する方法」の説明に従って、ソフトウェアベースのクライアントを構成できます。

次の例は、nxge NIC、ixgbe NIC、および e1000g NIC のリング関連のリングプロパティーに対して表示されるさまざまな情報を示しています。

例211 nxge NIC のリング情報

次の例は、nxge NIC のリング情報を示しています。

```bash
# dladm show-linkprop nxge

<table>
<thead>
<tr>
<th>LINK</th>
<th>PROPERTY</th>
<th>PERM</th>
<th>VALUE</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge0</td>
<td>rxrings</td>
<td>rw</td>
<td>--</td>
<td>sw, &lt;1-7&gt;</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge0</td>
<td>txrings</td>
<td>rw</td>
<td>--</td>
<td>sw, &lt;1-7&gt;</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge0</td>
<td>rxrings-available</td>
<td>r-</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge0</td>
<td>txrings-available</td>
<td>r-</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge0</td>
<td>rxhwclnt-available</td>
<td>r-</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge0</td>
<td>txhwclnt-available</td>
<td>r-</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

POSSIBLE フィールドには、rxrings および txrings プロパティーの受け入れ可能な値として sw と <1-7> が表示されます。これらの値は、nxge がソフトウェアベースのクライアントだけでなく、ハードウェアベースのクライアントをサポートしていることを示します。<1-7> の範囲は、設定する Rx リングまたは Tx リングの数が指定された範囲内なければならないことを示しています。また、NIC が受信側と送信側の両方で動的なリング割り当てをサポートしている範囲から推定することもできます。

さらに、*rings-available プロパティーは、ハードウェアベースのクライアントに割り当てのために 5 つ以上の Rx リングと 5 つの Tx リングが使用可能なことを示しています。

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
例21-1 nxge NICのリング情報（続き）

ただし、*clnt-availableプロパティに基づいて、使用可能なRxリングを排他的に使用できるクライアントは2つしか構成できません。同様に、使用可能なTxリングを排他的に使用できるクライアントは2つしか構成できません。

例21-2 ixgbe NICのリング情報

次の例は、ixgbe NICのリング情報を示しています。

| # dladm show-linkprop ixgbe0 LINK PROPERTY PERM VALUE DEFAULT POSSIBLE |
|-----------------|-----------------|---------|--------|-------------|
| ...             |                 |         |        |             |
| ixgbe0          | rxrings         | rw      | --     | --          | sw,hw   |
| ...             | txrings         | rw      | --     | --          | sw,hw,<1-7> |
| ixgbe0          | rxrings-available | r-     | 0      | --          |             |
| ixgbe0          | txrings-available | r-     | 5      | --          |             |
| ixgbe0          | rxhwclnt-available | r-     | 0      | --          |             |
| ixgbe0          | txhwclnt-available | r-     | 7      | --          |             |

rxringsプロパティとtxringsプロパティの両方のPOSSIBLEフィールドは、ixgbe0上にハードウェアベースのクライアントとソフトウェアベースのクライアントの両方を構成できることを示しています。Rxリングでは、静的なリング割り当てのみがサポートされています。つまり、ハードウェアがハードウェアベースの各クライアントに、固定されたRxリングのセットを割り当てます。ただし、Txリングは動的に割り当てることができます。つまり、ハードウェアベースのクライアントに割り当てるTxリングの数(この例では、最大7リング)を決定できます。

さらに、*rings-availableプロパティは、ハードウェアベースのクライアントに割り当てるために5つのTxリングが使用可能であるが、割り当て可能なRxリングはないと示しています。

最後に、*hwclnt-availableプロパティに基づいて、ハードウェアベースの7つのTxクライアントをTxリングを排他的に使用するように構成できます。ただし、ixgbeカードでは、動的なRxリング割り当てはサポートされていません。そのため、専用のRxリングの指定されたセットを備えたハードウェアベースのクライアントを作成することはできません。

どちらかの*rings-availableプロパティのVALUEフィールドにあるゼロ(0)は、次のどちらかを示すことがあります。

- クライアントに割り当てるために使用可能なリングがなくなった。
- 動的なリング割り当てがサポートされていない。

rxringsおよびtxringsのPOSSIBLEフィールドと、rxrings-availableおよびtxrings-availableのVALUEフィールドを比較することによって、この0の意味を確認できます。
データリンク上のリソースの管理

例21-2 ixgbe NICのリング情報（続き）

たとえば、次のように txrings-available が 0であるとします。

```bash
# dladm show-linkprop ixgbe0
LINK PROPERTY PERM VALUE DEFAULT POSSIBLE
...
ixgbe0 rxrings rw -- -- sw,hw
ixgbe0 txrings rw -- -- sw,hw,<1-7>
ixgbe0 txrings-available r- 0 -- --
ixgbe0 txrings-available r- 0 -- --
...
```

この出力では、rxrings-available の VALUE フィールドが 0であるのに対し
txrings の POSSIBLE フィールドは sw,hw です。これらの情報を組み合わせる
と、使用可能な Rx リングがないのは NIC が動的なリング割り当てをサポートしてい
ないためであることがわかります。送信側では、txrings-available の VALUE
フィールドが 0であるのに対して、txrings の POSSIBLE フィールドは sw,hw,<1-7> で
す。これらの情報を組み合わせると、使用可能な Tx リングがないのはすべての Tx リ
ングがすでに割り当てられているためであることがわかります。ただし、txrings
の POSSIBLE フィールドが示しているように、動的なリング割り当てはサポートされ
ています。そのため、Tx リングが使用可能になったら、これらのリングを割り当て
することができます。

例21-3 e1000g NICのリング情報

次の例は、e1000g NIC のリング情報を示しています。

```bash
# dladm show-linkprop e1000g0
LINK PROPERTY PERM VALUE DEFAULT POSSIBLE
...
e1000g0 rxrings rw -- --
e1000g0 txrings rw -- --
...
e1000g0 txrings-available r- 0 -- --
e1000g0 txrings-available r- 0 -- --
e1000g0 txhwclnt-available r- 0 -- --
e1000g0 txhwclnt-available r- 0 -- --
...
```

この出力は、e1000g NIC ではリング割り当てがサポートされていないため、リング
もハードウェアベースのクライアントも構成できないことを示します。

▼ ハードウェアベースのクライアントを構成する方法

この手順は、動的なリング割り当てをサポートする NIC、または静的なリング割り
当てをサポートする NIC 上でハードウェアベースのクライアントを構成する方法を
示しています。

始める前に システム上の NIC に関する次の情報を取得していることを確認してください。
データリンク上のリソースの管理

- NIC がハードウェアベースのクライアントをサポートしているかどうか
- NIC がサポートしているリング割り当てのタイプ
- ハードウェアベースのクライアントに割り当てためのリングの可用性
- リンク上に構成できるハードウェアベースのクライアントの可用性

これらの情報を取得するには、401 ページの「ハードウェアベースのクライアントを構成するための準備」を参照してください。

1. NIC がサポートしているリング割り当てのタイプに応じて、次の手順のいずれかを実行します。

   - NIC が動的なリング割り当てをサポートしている場合は、次の構文を使用します。

     ```
     # dladm create-vnic -p rxrings=number[,txrings=number] -l link vnic
     
     number  クライアントに割り当てる Rx リングと Tx リングの数を示します。この数は、割り当てに使用可能なリングの数の範囲内なければならない。
     ```

     注 - NIC によっては、Rx リングまたは Tx リングのどちらかで動的な割り当てをサポートしているが、両方のタイプではサポートしていないものがあります。number は、動的なリング割り当てがサポートされているリングのタイプに対して指定してください。

   - NIC が静的なリング割り当てをサポートしている場合は、次の構文を使用します。

     ```
     # dladm create-vnic -p rxrings=hw[,txrings=hw] -l link vnic
     
     hw  NIC によっては、Rx リングまたは Tx リングのどちらかで静的な割り当てをサポートしているが、両方のタイプではサポートしていないものがあります。hw は、静的なリング割り当てがサポートされているリングのタイプに対して指定してください。
     ```

2. (省略可能) 新しく作成されたクライアントのリング情報を確認します。

   ```
   # dladm show-linkprop vnic
   ```

▼ ソフトウェアベースのクライアントを作成する方法

ソフトウェアベースのクライアントは、リングを排他的に使用できません。代わりに、このクライアントは、主クライアントまたはインタフェースでのリングの使用
データリンク上のリソースの管理

を、ほかの既存のソフトウェアベースのクライアントと共有します。ソフトウェアベースのクライアントのリング数は、既存のハードウェアベースのクライアントの数によって異なります。

● 次の手順のいずれかを実行します。
  ■ 新しいソフトウェアベースのクライアントを作成するには、次のコマンドを入力します。

```
# dladm create-vnic -p rxrings=sw[,txrings=sw] -l link vnic
```

link クライアントの作成に使用しているデータリンクを示します。

vnic 構成しているクライアントを示します。
  ■ 既存のクライアントを、ほかのクライアントとリングを共有するように構成するには、次のコマンドを入力します。

```
# dladm set-linkprop -p rxrings=sw[,txrings=sw] vnic
```

例21-4 ハードウェアベースのクライアントとソフトウェアベースのクライアントの構成

この例は、ixgbe NICを備えたシステム上に、ハードウェアベースのクライアントとソフトウェアベースのクライアントの両方を構成する方法を示しています。リング割り当ての実装方法を示すために、この例はいくつかの部分に分けられています。構成プロセスの各手順で、リング関連の情報を表示され、説明されます。この構成は、次のように進められます。

1. クライアントの構成の前に、システム上のリンクとリング使用状況を表示します。
2. 主クライアントを構成します。
3. ソフトウェアベースのクライアントを構成します。
4. 専用リングの割り当てで別のクライアントを構成します。
5. 新しく構成されたクライアントにリングを静的に割り当てます。
6. 動的に割り当てられる専用リングを備えた3番目のクライアントを構成します。

最初に、リンク、リング使用状況、およびリング関連のプロパティを表示します。

```
# dladm show-link
LINK CLASS MTU STATE BRIDGE OVER
ixgbe0 phys 1500 down -- --
```

```
# dladm show-phys -H ixgbe0
LINK RINGTYPE RINGS CLIENTS
ixgbe0 RX 0-1 <default,mcast>
ixgbe0 TX 0-7 <default>
ixgbe0 RX 2-3 --
ixgbe0 RX 4-5 --
```

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
データリンク上のリソースの管理

```
ixgbe0  RX   6-7   --
# dladm show-linkprop ixgbe0
LINK  PROPERTY   PERM  VALUE  DEFAULT  POSSIBLE
... ixgbe0  rrxrings  rw  --  --  sw,hw
ixgbe0  rrxrings-effective  r  --  --  --
ixgbe0  txrxrings  rw  --  --  sw,hw,<1-7>
ixgbe0  txrxrings-effective  r  --  --  --
ixgbe0  txrxrings-available  r-  7  --  --
ixgbe0  rrxrings-available  r-  0  --  --
ixgbe0  rxhwclnt-available  r-  3  --  --
ixgbe0  txhwclnt-available  r-  7  --  --
...```

このコマンド出力はシステム上の1つのリンクixgbe0を示していますが、既存のクライアントは存在しません。さらに、この出力からは次の情報も収集されます。

- NICには8つのRxリングと8つのTxリング（リング0から7）があります。
- ハードウェアベースのクライアントの場合、Rxリングでは静的なリング割り当てのみがサポートされているのでに対して、Txリングでは静的なリング割り当てと動的なリング割り当ての両方がサポートされています。
- ソフトウェアベースのクライアントでは、RxリングとTxリングの両方を構成できます。
- 7つのTxリング（1から7）は、ほかのクライアントに動的に割り当てるために使用できます（リング0は通常、主クライアントのために予約されています）。Rxリングでは動的なリング割り当てがサポートされていないため、使用可能なRxリングはありません。
- Rxリングを使用するように構成できるハードウェアベースのクライアントが3つであるのに対して、Txリングを使用するように構成できるハードウェアベースのクライアントは7つです。

*rings-effectiveプロパティについては、412ページの「静的なリング割り当てにおけるリングの割り当てを識別する方法」を参照してください。

次に、主クライアントを構成します。

```
# ipadm create-ip ixgbe0
# ipadm create-addr -T static -a 192.168.10.10/24 ixgbe0/v4
# dladm show-phys -H ixgbe0
LINK  RINGTYPE  RINGS  CLIENTS
ixgbe0  RX  0-1  <default,mcast>
ixgbe0  TX  0-7  <default>ixgbe0
ixgbe0  RX  2-3  ixgbe0
ixgbe0  RX  4-5  --
ixgbe0  RX  6-7  --
# dladm show-linkprop ixgbe0
LINK  PROPERTY   PERM  VALUE  DEFAULT  POSSIBLE
...```
データリンク上のリソースの管理

<table>
<thead>
<tr>
<th>LINK</th>
<th>RINGTYPE</th>
<th>RINGS</th>
<th>CLIENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ixgbe0 RX</td>
<td>0-1</td>
<td></td>
<td>&lt;default,mcast&gt;,vnic0</td>
</tr>
<tr>
<td>ixgbe0 TX</td>
<td>0-7</td>
<td></td>
<td>&lt;default&gt;vnic0,ixgbe0</td>
</tr>
<tr>
<td>ixgbe0 RX</td>
<td>2-3</td>
<td></td>
<td>ixgbe0</td>
</tr>
<tr>
<td>ixgbe0 RX</td>
<td>4-5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ixgbe0 RX</td>
<td>6-7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

この出力では、次の情報が提供されます。

- 主クライアントであるixgbe0は、専用に使用するために2つのRxリング（リング2と3）を自動的に受け取ります。ただし、ixgbe0はすべてのTxリングを使用します。デフォルトでは、未使用のリングはすべて、主クライアントに自動的に割り当てられます。

- ほかのクライアントに割り当てることのできる使用可能なTxリングの数は7のままでです。

- Rxリングを構成できる使用可能なハードウェアベースのクライアントの数は3のままでです。Txリングを動的に構成できる使用可能なハードウェアベースのクライアントの数は7のままでです。

次に、ソフトウェアベースのクライアントとしてVNICを作成します。

```bash
# dladm create-vnic -l ixgbe0 -p rxrings=sw,txrings=sw vnic0
# dladm show-phys -H ixgbe0
LINK RINGTYPE RINGS CLIENTS
ixgbe0 RX 0-1 <default,mcast>,vnic0
ixgbe0 TX 0-7 <default>vnic0,ixgbe0
ixgbe0 RX 2-3 ixgbe0
ixgbe0 RX 4-5 --
ixgbe0 RX 6-7 --

# dladm show-linkprop vnic0
LINK PROPERTY PERM VALUE DEFAULT POSSIBLE
... vnic0 rxrings rw sw -- sw,hw
... vnic0 txrings rw sw -- sw,hw,<1-7>

# dladm show-linkprop ixgbe0
LINK PROPERTY PERM VALUE DEFAULT POSSIBLE
... ixgbe0 rxrings rw -- -- --
ixgbe0 rxrings-effective r 2 -- --
ixgbe0 txrings rw -- -- sw,hw,<1-7>
ixgbe0 txrings-effective r -- -- --
ixgbe0 txrings-available r- 7 -- --
ixgbe0 rxrings-available r- 0 -- --
ixgbe0 rxhwclnt-available r- 3 -- --
ixgbe0 txhwclnt-available r- 7 -- --
...```
この出力では、次の情報が提供されます。

- ソフトウェアベースのクライアントとして、vnic0 には、Rx リング 0 と 1 の使用が自動的に割り当てられます。その後に作成される、Rx リングを構成した他のソフトウェアベースのクライアントには、デフォルトで、このベンの使用が割り当てられます。デフォルトでは、vnic0 には、8 のすべての Tx リング（リング 0 から 7）の使用も割り当てられます。その後に作成される、Tx リングを構成した他のソフトウェアベースのクライアントには、デフォルトで、このリングのセットの使用が割り当てられます。
- ソフトウェアベースのクライアントとして、vnic0 の rxrings および txrings プロパティは、それに応じて sw に設定されます。
- Tx リングは割り当てられません。その結果、ほかのクライアントに割り当てることのできる使用可能な Tx リングの数は 7 のままです。
- Rx リングは構成できる使用可能なハードウェアベースのクライアントの数は 3 のままです。Tx リングを構成できる使用可能なハードウェアベースのクライアントの数は 7 のままです。

次に、リング割り当てなしで別のクライアントを構成します。

```
# dladm create-vnic -l ixgbe0 vnic1
# dladm show-phys -H ixgbe0
<table>
<thead>
<tr>
<th>LINK</th>
<th>RINGTYPE</th>
<th>RINGS</th>
<th>CLIENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ixgbe0</td>
<td>RX</td>
<td>0-1</td>
<td>&lt;default,mcast&gt;,vnic0</td>
</tr>
<tr>
<td>ixgbe0</td>
<td>TX</td>
<td>0,2-7</td>
<td>&lt;default&gt;vnic0,ixgbe0</td>
</tr>
<tr>
<td>ixgbe0</td>
<td>RX</td>
<td>2-3</td>
<td>ixgbe0</td>
</tr>
<tr>
<td>ixgbe0</td>
<td>RX</td>
<td>4-5</td>
<td>vnic1</td>
</tr>
<tr>
<td>ixgbe0</td>
<td>RX</td>
<td>6-7</td>
<td>--</td>
</tr>
<tr>
<td>ixgbe0</td>
<td>TX</td>
<td>1</td>
<td>vnic1</td>
</tr>
</tbody>
</table>
```

```
# dladm show-linkprop vnic1
<table>
<thead>
<tr>
<th>LINK</th>
<th>PROPERTY</th>
<th>PERM</th>
<th>VALUE</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnic1</td>
<td>rxrings</td>
<td>rw</td>
<td>--</td>
<td>--</td>
<td>sw,hw</td>
</tr>
<tr>
<td>vnic1</td>
<td>rxrings-effective</td>
<td>r-</td>
<td>2</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>vnic1</td>
<td>txrings</td>
<td>rw</td>
<td>--</td>
<td>--</td>
<td>sw,hw,&lt;1-7&gt;</td>
</tr>
<tr>
<td>vnic1</td>
<td>txrings-effective</td>
<td>r-</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
```

```
# dladm show-linkprop ixgbe0
<table>
<thead>
<tr>
<th>LINK</th>
<th>PROPERTY</th>
<th>PERM</th>
<th>VALUE</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ixgbe0</td>
<td>rxrings</td>
<td>rw</td>
<td>--</td>
<td>--</td>
<td>sw,hw</td>
</tr>
<tr>
<td>ixgbe0</td>
<td>rxrings-effective</td>
<td>r-</td>
<td>2</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>ixgbe0</td>
<td>txrings</td>
<td>rw</td>
<td>--</td>
<td>--</td>
<td>sw,hw,&lt;1-7&gt;</td>
</tr>
<tr>
<td>ixgbe0</td>
<td>txrings-available</td>
<td>r-</td>
<td>7</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>ixgbe0</td>
<td>rxhwclnt-available</td>
<td>r-</td>
<td>0</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>ixgbe0</td>
<td>txhwclnt-available</td>
<td>r-</td>
<td>3</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>ixgbe0</td>
<td>txhwclnt-available</td>
<td>r-</td>
<td>7</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
```

データリンク上のリソースの管理

第21章・ネットワークリソースの管理 409
データリンク上のリソースの管理

この出力では、次の情報が提供されます。

- リング割り当てがサポートされている場合、構成されているクライアントは、rxringsおよびtxringsプロパティが設定されていなくてもハードウェアベースのクライアントとみなされます。そのため、vnic1は、自ら使用するためには2つの専用のRxリング（リング4と5）を自動的に受け取ります。同様に、vnic1も専用のTxリング（リング1）を受け取ります。
- 8つのTxリングのうち、ixgbe0とvnic0は現在7つのリング（リング0とリング2から7）を共有しています。リング1は、vnic1の専用のTxリングになっています。
- Txリングは割り当てられません。そのため、ほかのクライアントに割り当てることのできる使用可能なTxリングの数は7のままでです。
- Rxリングを構成できる使用可能なハードウェアベースのクライアントの数は3のままでです。Txリングを構成できる使用可能なハードウェアベースのクライアントの数は7のままでです。

次に、新しく構成されたクライアントvnic1にリングを静的に割り当てます。

```
# dladm set-linkprop -p rxrings=hw,txrings=hw vnic1
# dladm show-phys -H ixgbe0
LINK     RINGTYPE  RINGS     CLIENTS
ixgbe0   RX        0-1       <default,mcast>,vnic0
ixgbe0   TX        0,2-7    <default>,vnic0,ixgbe0
ixgbe0   RX        2-3      ixgbe0
ixgbe0   RX        4-5      vnic1
ixgbe0   RX        6-7      --
ixgbe0   TX        1        vnic1
```

```
# dladm show-linkprop vnic1
LINK       PROPERTY     PERM  VALUE  DEFAULT  POSSIBLE
          ...           ...  ...    ...      ...
ixgbe0    rxrings      rw    hw    --        sw,hw
ixgbe0    rxrings-effective r-  2    --       --
ixgbe0    txrings      rw    hw    --        sw,hw,<1-7>
ixgbe0    txrings-effective r-  --    --       --
ixgbe0    txhwclnt-available r-  3        --      --
ixgbe0    rnxhwclnt-available r-  6        --      --
```

```
# dladm show-linkprop ixgbe0
LINK       PROPERTY     PERM  VALUE  DEFAULT  POSSIBLE
          ...           ...  ...    ...      ...
ixgbe0    rxrings      rw    --    --        sw,hw
ixgbe0    rxrings-effective r-  2    --       --
ixgbe0    txrings      rw    --    --        sw,hw,<1-7>
ixgbe0    txrings-available r-  --    --       --
ixgbe0    rnxhwclnt-available r-  3        --      --
ixgbe0    txhwclnt-available r-  6        --      --
```

この出力では、次の情報が提供されます。

Oracle Solaris管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
vnic1でのRxリングとTxリングの分布は、vnic1のリング割り当てなしで作成されたときと同じです。

同様に、リング情報も、vnic1がリング割り当てなしで作成されたときと同じです。

vnic1のrxringsおよびtxringsプロパティは、明示的にhwに設定されています。その結果、動的な割り当てに使用可能なTxリングの数は6に減っています。同様に、構成できる使用可能なハードウェアベースのクライアントの数も6に減っています。

次に、動的に割り当てられるTxリングを備えたハードウェアベースのクライアントを構成します。

```
# dladm create-vnic -l ixgbe0 -p txrings=2 vnic2
# dladm show-phys -H ixgbe0
LINK  RINGTYPE  RINGS  CLIENTS
ixgbe0 RX 0-1 <default,mcast>,vnic0
ixgbe0 TX 0,4-7 <default>vnic0,ixgbe0
ixgbe0 RX 2-3 ixgbe0
ixgbe0 RX 4-5 vnic1
ixgbe0 RX 6-7 vnic2
ixgbe0 TX 1 vnic1
ixgbe0 TX 2-3 vnic2
```

```
# dladm show-linkprop vnic2
LINK PROPERTY PERM VALUE DEFAULT POSSIBLE
... vnic2 rxrings rw -- -- sw,hw
vnic2 rxrings-effective r- 2 -- --
... vnic2 txrings rw 2 -- sw,hw,<1-7>
vnic2 txrings-effective r- 2 -- --
```

```
# dladm show-linkprop ixgbe0
LINK PROPERTY PERM VALUE DEFAULT POSSIBLE
... ixgbe0 rxrings rw -- -- sw,hw
ixgbe0 rxrings-effective r- 2 -- --
ixgbe0 txrings rw -- --
ixgbe0 txrings-effective r- -- --
ixgbe0 txrings-available r- 4 -- --
ixgbe0 rxwclnt-available r- 3 -- --
ixgbe0 txwclnt-available r- 5 -- --
```

この出力では、次の情報が提供されます。

ハードウェアは、vnic2にRxリングのペア(リング6と7)を排他的使用のために自動的に割り当てました。ただし、vnic2の2つの専用のTxリング(リング2と3)は管理者によって割り当てられました。

2つのTxリングが管理用にvnic2に割り当てられているため、ほかのクライアントに割り当てることのできる使用可能なTxリングの数は4に減っています。
データリンク上のリソースの管理

- vnic2が2つのTxリングを備えたハードウェアベースのクライアントとして構成されているため、構成できる使用可能なクライアントの数は5に減っています。

▼ 静的なリング割り当てにおけるリングの割り当てを識別する方法
静的なリング割り当てでハードウェアベースのクライアントを構成すると、割り当てられるリングの数がハードウェアによって決定されます。ただし、rxringsおよびtxringsプロパティはhwに設定されていて、実際に割り当てられているリングの数を示していません。代わりに、rxrings-effectiveおよびtxrings-effectiveプロパティを確認することによって、この数を取得できます。

1. 次の手順のいずれかを実行することによって、静的なリング割り当てでハードウェアベースのクライアントを構成します。

- 静的なリング割り当てでクライアントを作成するには、次のコマンドを入力します。
  ```
  # dladm create-vnic -l link -p rxrings=hw[,txrings=hw] vnic
  link クライアントの作成に使用しているデータリンクを示します。
  vnic 構成しているクライアントを示します。
  # dladm set-linkprop -p rxrings=hw[,txrings=hw] vnic
  既存のクライアントにリングを静的に割り当てるには、次のコマンドを入力します。
  # dladm set-linkprop -p rxrings=hw[,txrings=hw] vnic
  
 2. 割り当てられているリングの数を識別するには、次のサブステップを実行します。

  a. クライアントのプロパティを表示します。
     ```
     # dladm show-linkprop link
     ここで、linkは、ハードウェアベースのクライアントまたはVNICを示します。
     ```

  b. 静的に割り当てたリングのタイプに対応する*rings-effectiveプロパティの値を確認します。
     たとえば、Rxリングを静的に割り当たくなった場合は、rxrings-effectiveプロパティを確認します。Txリングを静的に割り当たった場合は、txrings-effectiveプロパティを確認します。この数は、ハードウェアによって割り当てられたリングの数を示します。

3. どのリングが静的に割り当てられているかを確認するには、次のサブステップを実行します。

  a. NICのリング使用状況を表示します。
     ```
     # dladm show-phys -H link
     ```
ここで、linkは主クライアントを示します。

b. コマンド出力から、最初の手順で構成したハードウェアベースのクライアントに、どのRxリングまたはTxリングが割り当てられているかを確認します。

例 21-5 静的に割り当てられているリングの識別

この例は、ixgbe NIC上に構成されているクライアントにRxリングがどのように静的に割り当てられているかを示しています。このようなNICでは、Rxリングには静的な割り当てのみがサポートされています。この例は、次のように進められます。

1. システム上のリンクを表示します。この例では、システムには1つのリンク(ixgbe0)しかありません。
2. 静的に割り当てられるRxリングを備えたハードウェアベースのクライアントとしてvnic1を作成します。
3. リング情報を表示して、ハードウェアによって割り当てられたリングの数を確認します。
4. リング使用状況を表示して、どのリングが割り当てられているかを識別します。

```bash
# dladm show-link
LINK    CLASS  MTU  STATE  BRIDGE  OVER
ixgbe0  phys  1500  down  --  --

# dladm create-vnic -l ixgbe0 -p rxrings=hw vnic1
# dladm show-linkprop vnic1
LINK    PROPERTY     PERM  VALUE  DEFAULT  POSSIBLE
... 
vnic1  rxrings       rw   hw   --       sw,hw
vnic1  rxrings-effective  r-  2   --   --
vnic1  txrings        rw   --   --       sw,hw,<1-7>
vnic1  txrings-effective  r-  --   --   --

# dladm show-phys -H ixgbe0
LINK    RINGTYPE  RINGS  CLIENTS
ixgbe0  RX   0-1     <default,mcast>
ixgbe0  TX   0,2-7   <default>
ixgbe0  RX   2-3     vnic1
ixgbe0  RX   4-5     --
ixgbe0  RX   6-7     --
ixgbe0  TX   1       vnic1
...
```

この出力は、rxrings-effectiveプロパティに反映されているように、vnic1にRxリングが構成されたあと、ハードウェアによって2つの専用のRxリングが割り当てられたことを示します。dladm show-phys -H コマンドの出力に基づくと、Rxリング2と3がvnic1で専用に使用するために割り当てられました。
クライアントとして構成された結果として、vnict1 はまた、専用で使用するために
 Tx リンクも自動的に受け取りました。ただし、txrings プロパティーが明示的に設
定されていないため、txrings-effective プロパティーには値が表示されていません。

プールと CPU

pool は、ネットワーク処理を CPU のプールにバインドできるようにするためのリン
クプロパティーです。このプロパティーを使用すると、ネットワークリソースの管
理を、ゾーン内の CPU バインディングや管理より適切に統合できます。Oracle
Solaris でのゾーン管理には、zonecfg または poolcfg コマンドを使用して、ネット
ワーク以外のプロセスを CPU リソースのプールにバインドする処理が含まれま
す。その同じリソースのプールをネットワークプロセスも管理するようにバインド
するには、dladm set-linkprop コマンドを使用してリンクの pool プロパティーを構
成します。次に、ゾーンにそのリンクを割り当てます。

リンクの pool プロパティーを設定し、そのリンクをゾーンのネットワークインタ
フェースとして割り当てることによって、そのリンクがゾーンのプールにもバイン
ドされます。そのゾーンが排他的ゾーンになるように設定されると、プール内の
CPU リソースは、そのゾーンに割り当てられていないほかのデータリンクから使用
できなくなります。

注- 別のプロパティー cpu を設定すると、データリンクに特定の CPU を割り当てるこ
とができます。この 2 つのプロパティー cpu と pool は相互に排他的です。特定の
データリンク用に両方のプロパティーを設定することはできません。cpu プロパ	
テーを使用してデータリンクに CPU リソースを割り当てるには、418 ページの「リン	
クに CPU を割り当てる方法」を参照してください。

ゾーン内のプールについての詳細は、『Oracle Solaris のシステム管理 (Oracle Solaris
ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』の第 13 章「リソースプール
の作成と管理 (手順)」を参照してください。プールの作成および CPU セットの
プールへの割り当てについての詳細は、poolcfg(1M) のマニュアルページを参照して
ください。

次の図は、データリンクに pool プロパティーが割り当てられたときのプールの動作
を示しています。
この図では、システムに8つのCPUがあります。システム上にプールが構成されていない場合は、すべてのCPUがデフォルトプールに属し、大域ゾーンで使用されます。ただし、この例では、CPU3とCPU4で構成されたpool99プールが作成されています。このプールは、排他的ゾーンであるzone1に関連付けられています。pool99がvnic1のプロパティとして設定された場合、pool99はvnic1のネットワークプロセスの管理に専用に使用されます。vnic1がzone1のネットワークインタフェースとして割り当てられたあと、pool99内のCPUは、zone1のネットワークプロセスとネットワーク以外のプロセスの両方を管理するために予約されます。

poolプロパティは本質的に動的です。ゾーンのプールには一連のCPUを構成することができ、そのプールのCPUセットにどのCPUが割り当てられるかはカーネルによって決定されます。プールに対する変更はデータリンク用に自動的に実装されるため、そのリンク用のプールの管理が単純化されます。これに対して、cpuプロパティを使用してリンクに特定のCPUを割り当てるには、割り当てられるCPUを指定する必要があります。プールのCPUコンポーネントを変更しようとするたびにcpuプロパティを設定する必要があります。

たとえば、図21-1のシステムでCPU4がオフラインになったとします。poolプロパティは動的であるため、ソフトウェアによって追加のCPUがプールに自動的に関連付けられます。そのため、2CPUというプールの元の構成が保持されます。vnic1にとっては、この変更は透過的です。調整された構成を次の図に示します。
プール関連の追加のプロパティによって、CPU または CPU のプールのデータリンクの使用に関する情報が表示されます。これらのプロパティは読み取り専用であり、管理者が設定することはできません。

- pool-effective は、ネットワークプロセスに使用されているプールを表示します。
- cpus-effective は、ネットワークプロセスに使用されている CPU の一覧を表示します。

ゾーンの CPU リソースを管理するために、初期手順として、データリンクの pool プロパティが設定されるのは通常ありません。より一般的な方法として、リソースのプールを使用するゾーンを構成するために zonecfg や poolcfg などのコマンドが使用されます。cpu や pool のリンクプロパティ自体が設定されることはありません。その場合は、これらのデータリンクの pool-effective および cpus-effective プロパティが、ゾーンがプートされたときのこれらのゾーン構成に従って自動的に設定されます。pool-effective ではデフォルトプールが表示されるのに対して、cpus-effective の値はシステムによって選択されます。そのため、dladm show-linkprop コマンドを使用した場合、pool-effective および cpus-effective プロパティには値が含まれていますが、pool および cpu プロパティは空です。

データリンクの pool および cpu プロパティの直接の設定は、ゾーンの CPU プールをネットワークプロセス用にバインドするために使用できる代替手順です。これらのプロパティを構成すると、それらの値が pool-effective および cpus-effective プロパティに反映されます。ただし、この代替手順を使用してゾーンのネットワークリソースを管理することは通常、あまり行われません。

▼ CPU プールをデータリンク用に構成する方法

ほかのリンクプロパティと同様に、pool プロパティも、そのリンクが作成された時点でか、またはあとでそのリンクをさらに構成する必要が発生したときにデータリンク用に設定できます。たとえば:
データリンク上のリソースの管理

# dladm create-vnic -p pool=pool-name -l link vnic

VNICの作成中にpoolプロパティを設定します。既存のVNICのpoolプロパティを設定するには、次の構文を使用します。

# dladm setlinkprop -p pool=pool-name vnic

CPUプールをVNIC用に構成するための手順を次に示します。

始める前に
次の作業を完了している必要があります。
- 割り当てられた数のCPUを含むプロセッサセットを作成します。
- そのプロセッサセットを関連付けるプールを作成します。
- そのプールをプロセッサセットに関連付けます。

注 - これらの前提条件を完了するための手順については、『Oracle Solarisのシステム管理(Oracle Solarisゾーン、Oracle Solaris 10ゾーン、およびリソース管理)』の「構成の変更方法」を参照してください。

1 リンクのpoolプロパティを、ゾーン用に作成したCPUのプールに設定します。VNICが存在するかどうかに応じて、次の手順のいずれかを実行します。
- VNICがまだ作成されていない場合は、次の構文を使用します。
  # dladm create-vnic -l link -p pool=pool vnic
  ここで、poolは、ゾーン用に作成されたプールの名前を示します。
- VNICが存在する場合は、次の構文を使用します。
  # dladm setlinkprop -p pool=pool vnic

2 ゾーンをそのVNICを使用するように設定します。
  zonecfg=zonedid:net> set physical=vnic

注 - ゾーンにネットワークインタフェースを割り当てる方法を説明した完全な手順については、『Oracle Solarisのシステム管理(Oracle Solarisゾーン、Oracle Solaris 10ゾーン、およびリソース管理)』の「ゾーンを構成、検証、および確定する」を参照してください。

例 21-6 リンクのCPUプールを排他的なIPタイプを持つゾーンに割り当てる

この例では、プールをゾーンのデータリンクに割り当てる方法を示しています。このシナリオは、図21-1の構成に基づいています。この例では、pool99という名前のCPUのプールが、すでにゾーン用に構成されていることを前提にしています。この
データリンク上のリソースの管理

プールが次に、VNICに割り当てられます。最後に、非大域ゾーンzone1が、このVNICをネットワークインタフェースとして使用するように設定されます。

```
# dladm create-vnic -l e1000g0 -p pool99 vnic0
# zonecfg -c zone1
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1> add net
zonecfg:zone1>net> set physical=vnic0
zonecfg:zone1>net> end
zonecfg:zone1> exit
```

▼ リンクにCPUを割り当てる方法
次的手順は、cpuプロバティーを構成することによって、データリンクをたどってい
るトラフィックの処理に特定のCPUを割り当てる方法を説明しています。

1 インタフェースのCPU割り当てを確認します。
```
# dladm show-linkprop -p cpus link
```
デフォルトでは、特定のどのインタフェースにもCPUは割り当てられていま
せん。そのため、コマンド出力にあるパラメータVALUEにはどのエントリも含まれませ
ん。

2 各割り込みと、それらの割り込みが関連付けられているCPUを一覧表示します。
```
# echo ::interrupts | mdb -k
```
この出力には、システム内のリンクごとのパラメータ(CPU番号を含む)が一覧表示
されます。

3 リンクにCPUを割り当てます。
このCPUには、そのリンクの割り込みが関連付けられているCPUを含めることがで
きます。
```
# dladm set-linkprop -p cpus=cpu1,cpu2,... link
```
ここで、cpu1は、そのリンクに割り当てるCPU番号です。そのリンク専用に複数の
CPUを割り当てることができます。

4 リンクの割り込みをチェックして、新しいCPU割り当てを確認します。
```
# echo ::interrupts | mdb -k
```

5 (省略可能)リンクに関連付けられたCPUを表示します。
```
# dladm show-linkprop -p cpus link
```
例21-7 インタフェースへのCPUの割り当て
この例は、図18-3のinternal0インタフェース専用に特定のCPUを割り当てる方法を示しています。

さまざまなコマンドで生成される出力にある次の情報に注目してください。わかりやすくするために、出力にある重要な情報が強調されています。

- デフォルトでは、internal0に専用のCPUは割り当てられていません。そのため、VALUEは--です。
- internal0の割り込みは、CPU18に関連付けられています。
- CPUが割り当てられたあと、internal0のVALUEには新しいCPUの一覧が表示されます。

```bash
# dladm show-linkprop -p cpus internal0
<table>
<thead>
<tr>
<th>LINK</th>
<th>PROPERTY</th>
<th>PERM</th>
<th>VALUE</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>internal0</td>
<td>cpus</td>
<td>rw</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

# echo ::interrupts | mdb -k
Device Shared Type MSG # State IND Mondo Pl  CPU
external#0 no MSI 3 enbl 0x1b 0x1b 6 0
internal#0 no MSI 2 enbl 0x1a 0x1a 6 18

# dladm set-linkprop -p cpus=14,18,19,20 internal0

# dladm show-linkprop -p cpus internal0
<table>
<thead>
<tr>
<th>LINK</th>
<th>PROPERTY</th>
<th>PERM</th>
<th>VALUE</th>
<th>DEFAULT</th>
<th>POSSIBLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>internal0</td>
<td>cpus</td>
<td>rw</td>
<td>14,18,19,20</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
```

割り込みを含め、サポートしているスレッドはすべて、新しく割り当てられた一連のCPUに限定されるようになりました。

フローラ上のリソースの管理

フローは、属性に従って構成されたネットワークパケットから成ります。フローを使用すると、ネットワークリソースをさらに詳細に割り当てることができます。フローの概要については、396ページの「フローを使用することによるネットワークリソースの管理」を参照してください。

リソースの管理にフローを使用するには、次のような一般的な手順を実行します。

1. 396ページの「フローを使用することによるネットワークリソースの管理」に示されている特定の属性に基づいてフローを作成します。
2. ネットワークリソースに関連するプロパティを設定することによって、リソースのフローの使用をカスタマイズします。現在、設定できるのはパケットを処理するための帯域幅だけです。
ネットワーク上のフローの構成

フローは、物理ネットワークだけでなく仮想ネットワーク上にも作成できます。フローを構成するには、flowadmコマンドを使用します。詳細な技術情報については、flowadm(1M)のマニュアルページを参照してください。

▼ フローを構成する方法

1 (省略可能) フローを構成するリンクを決定します。
   # dladm show-link

2 選択されたリンク上のIPインタフェースにIPアドレスが正しく構成されていることを確認します。
   # ipadm show-addr

3 フローごとに指定した属性に従ってフローを作成します。
   # flowadm add-flow -l link -a attribute=value[,attribute=value] flow
   attribute ネットワークパケットをフローに構成するときに使用できる次の分類のいずれかを示します。
   - IPアドレス
   - トランスポートプロトコル(UDP、TCP、またはSCTP)
   - アプリケーションのポート番号(たとえば、FTPではポート21)
   - DSフィールドの属性(IPv6パケットでのサービスの品質にのみ使用されます)DSフィールドについての詳細は、「Oracle Solarisの管理:IPサービス」の「DSコードポイント」を参照してください。

   flow 特定のフローに割り当てる名前を示します。
   フローとフローの属性についての詳細は、flowadm(1M)のマニュアルページを参照してください。

4 該当するフローのプロバティーを設定することによって、フロー上にリソース制御を実装します。
   # flowadm set-flowprop -p property=value[,property=value,...] flow
   リソースを制御するための次のフローのプロバティーを指定できます。

   maxbw このフローで識別されたパケットが使用できるリンクの帯域幅の最大量。設定する値は、リンクの帯域幅に対して許可される値の範囲内なければなりません。リンクの帯域幅に対して指定可能な値の範囲を表示するには、次のコマンドで生成される出力にあるPOSSIBLEフィールドを確認します。
フロー上のリソースの管理

# dladm show-linkprop -p maxbw link

注 - 現在、カスタマイズできるのはフローの帯域幅だけです。

5 (省略可能) リンク上に作成したフローを表示します。
   # flowadm show-flow -l link

6 (省略可能) 指定されたフローのプロパティ設定を表示します。
   # flowadm show-flowprop flow

例 21-8 リンクとフローのプロパティを設定することによるリソースの管理

この例では、データリンクとフローの両方にネットワークリソースを割り当てるための手順が結合されています。この例は、次の図に示す構成に基づいています。
フロー上のリソースの管理

この図は、互いに接続された2つの物理ホストを示しています。

- **Host1**の構成は次のとおりです。
  - ルーターゾーンとして機能する1つの非大域ゾーンが存在します。このゾーンには、次の2つのインタフェースが割り当てられています。`external0`がインターネットに接続するのに対して、`internal0`は2番目のホストを含む内部ネットワークに接続します。
  - IPインタフェースは、カスタマイズされた名前を使用するように名前が変更されています。必須ではありませんが、リンクやインタフェースにカスタマイズされた名前を使用すると、ネットワークを管理するときに利点が得られます。26ページの「ネットワークデバイスとデータリング名」を参照してください。
フロー上のリソースの管理

- UDPトラフィックを分離し、UDPパケットのリソースの使用方法に対する制御を実装するために、internal0上にフローが構成されています。フローの構成については、419ページの「フロー上のリソースの管理」を参照してください。

- Host2の構成は次のとおりです。
  - 3つの非大域ゾーンと、それぞれ対応するVNICが存在します。これらのVNICは、動的なリンク割り当てをサポートするnxgeカード上に構成されています。リンク割り当てについての詳細は、399ページの「送信リンクと受信リンク」を参照してください。
  - 各ゾーンのネットワーク処理の負荷は異なります。この例の目的として、zone1の負荷は高く、zone2の負荷は中程度であり、zone3の負荷は軽くなっています。リソースは、それぞれの負荷に従ってこれらのゾーンに割り当てられています。
  - 別のVNICがソフトウェアベースのクライアントとして構成されていま
  す。MACクライアントの概要については、399ページの「MACクライアントとリング割り当て」を参照してください。

この例でのタスクには、次のものが含まれます。

- フローの作成およびフロー制御の構成 Host2で受信されるUDPパケットに対する個別のリソース制御を作成するために、internal0上にフローが作成されます。

- Host2上のVNICのためのネットワーククリソースのプロバティーの構成 - 各ゾーン上の処理の負荷に基づいて、各ゾーンのVNICに一連の専用リンクが構成されます。また、ソフトウェアベースのクライアントの例として、専用リンクのない別のVNICも構成されます。

この例には、ゾーン構成のための手順は含まれていないことに注意してください。ゾーンを構成するには、『Oracle Solarisのシステム管理(Oracle Solarisゾーン、Oracle Solaris 10ゾーン、およびリソース管理)』の第17章「非大域ゾーンの計画と構成(手順)」を参照してください。

最初に、Host1上のリンクとIPインタフェースに関する情報を表示します。

```
# dladm show-phys
LINK MEDIA STATE SPEED DUPLEX DEVICE
internal0 Ethernet up 1000 full nge1
e1000g0 n unknown 0 half e1000g0
e1000g1 n unknown 0 half e1000g1
external0 Ethernet up 1000 full nge0
```

```
# dladm show-link
LINK CLASS MTU STATE BRIDGE OVER
internal0 phys 1500 up -- nge1
e1000g0 phys 1500 unknown -- --
e1000g1 phys 1500 unknown -- --
```
フロー上のリソースの管理

```
external0  phys  1500  up   --  nge0

# ipadm show-addr
ADDROBJ  TYPE  STATE  ADDR
lo0/4    static  ok   127.0.0.1/8
external0 static  ok   10.10.6.5/24
internal0 static  ok   10.10.12.42/24
```

次に、Host2へのUDPトラフィックを分離するために、internal0上にフローを作成します。次に、そのフロー上にリソース制御を実装します。

```
# flowadm add-flow -l external0 -a transport=udp udpflow
# flowadm set-flowprop -p maxbw=80 udpflow
```

次に、作成されたフローに関する情報を確認します。

```
flowadm show-flow
FLOW  LINK  IPADDR  PROTO  PORT  DFSLD
udpflow  internal0  --  udp  --  --

# flowadm show-flowprop
SECURE OUTPUT FOR THIS
```

Host2上で、ゾーンごとにnxge0上にVNICを構成します。各VNIC上にリソース制御を実装します。次に、各VNICをそれぞれ対応するゾーンに割り当てます。

```
# dladm create-vnic -l nxge0 vnic0
# dladm create-vnic -l nxge0 vnic1
# dladm create-vnic -l nxge0 vnic2

# dladm set-prop -p rxrings=4,txrings=4 vnic0
# dladm set-prop -p rxrings=2,txrings=2 vnic1
# dladm set-prop -p rxrings=1,txrings=1 vnic2

# zone1>zonecfg>net>
zone1>zonecfg>net> set physical=vnic0
zone2>zonecfg>net> set physical=vnic1
zone3>zonecfg>net> set physical=vnic2
```

Host2内の1台のCPUであるpool1が、以前にzone1で使用するように構成されていました。次のように、そのCPUのプールをzone1のネットワークプロセスも管理するようにバインドします。

```
# dladm set-prop -p pool=pool01 vnic0
```

最後に、主インタフェースであるnxge0とリングを共有するソフトウェアベースのクライアントを作成します。

```
dladm create-vnic -p rxrings=sw,txrings=sw -l nxge0 vnic3
```
第 22 章

ネットワークトラフィックとリソース使用状況の監視

この章では、物理および仮想ネットワーク環境でのネットワークリソースの使用に関す続情報を監視したり、収集したりするためのタスクについて説明します。これらの情報は、プロビジョニング、統合、および請求の目的でリソース割り当てを解析するために役立ちます。この章では、dlstat と flowstat という、統計情報表示するために使用する 2 つのコマンドについて説明します。

この章では、次の内容について説明します。
- 425 ページの「ネットワークトラフィックフローの概要」
- 428 ページの「トラフィックとリソース使用の監視（タスクマップ）」
- 429 ページの「リンク上のネットワークトラフィックに関する統計情報の収集」
- 435 ページの「プローグ上のネットワークトラフィックに関する統計情報の収集」
- 437 ページの「ネットワークアカウントでの設定」

ネットワークトラフィックフローの概要

パケットは、システムに入力されるか、またはシステムから出力されるときにパスをたどります。詳細なレベルで見ると、パケットは、NIC の受信 (Rx) リングおよび送信 (Tx) リングを通じて送受信されます。これらのリングから、受信パケットはさらに処理するためにネットワークスキャッドに渡され、送信パケットはネットワークに送信されます。

第 21 章「ネットワークリソースの管理」では、ネットワーククレーンの概念について説明します。ネットワークトラフィックを管理するために割り当てられたシステムリソースの結合法によって、ネットワーククレーンが構成されます。そのため、ネットワーククレーンは、特定のタイプのネットワークトラフィックのためにカスタマイズされたパスです。各レーンは、ハードウェアレーンまたはソフトウェアレーンのどちらかにすることができます。さらに、各レーンのタイプを受信レーンまたは送信レーンのどちらかにすることができます。ハードウェアレーンとソフトウェアレーンの区別は、リング割り当てをサポートする NIC の能力に基づいています。リ
ネットワークトラフィックフローの概要

リング割り当ての詳細は、399ページの「送信リングと受信リング」を参照してください。この章では主に、受信レーンを通して受信される受信トラフィックについて重点的に説明します。

ハードウェアレーン上では、リングは、それらのレーンを使用するパケット専用になります。これに対して、ソフトウェアレーン上のリングはデータリンクの間で共有されます。データリンクは、次の理由から、リングを共有するように構成されます。

■ 管理の目的。データリンクが、専用リングを必要とするリソース集約型プロセスを実行していない可能性があります。
■ NICでリング割り当てがサポートされていない。
■ リング割り当てがサポートされているにもかかわらず、リングを他の使用のために割り当てることができなくなった。

さまざまなハードウェアレーンを示す次の図について検討します。
この図は、次の構成を示しています。

- システムには、nxge という 1 枚の NIC があります。
トラフィックとリソース使用の監視（タスクマップ）

- リンクは、物理デバイス nxge0、vnic1、および vnic2 上に構成されています。データリンクとして、nxge0 には、カスタマイズされた名前を割り当てることができます。ただし、この図で、リンクはデフォルトのデバイス名を保持しています。
- システムには複数の CPU があります。
- この NIC は、動的なリング割り当てをサポートしています。そのため、ハードウェアレーンを構成するために、各リンクに一連のハードウェアリングを割り当てることができます。さらに、各レーンには一連の CPU も割り当てられます。

トラフィックとリソース使用の監視（タスクマップ）

ネットワークレーン上のパケットフローを監視することによって、パケットによってネットワークリソースがどのように使用されているかに関する情報を取得できます。dstat コマンドでは、データリンクに関するこれらの情報が提供されます。flowstat コマンドは、既存のフローに対して同様の機能を実行します。

次の表は、システム内のネットワークトラフィックとリソース使用に関する統計情報を取得するために使用できる各種の方法を示しています。

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>ネットワークトラフィックに関する統計情報を取得します。</td>
<td>システムのネットワークインタフェース上の受信および送信トラフィックを表示します。</td>
<td>430 ページの「ネットワークトラフィックに関する基本的な統計情報を取得する方法」</td>
</tr>
<tr>
<td>リンクの使用に関する統計情報を取得します。</td>
<td>受信および送信トラフィックが NIC のリング間でどのように分散されているかを表示します。</td>
<td>432 ページの「リング使用状況に関する統計情報を取得する方法」</td>
</tr>
<tr>
<td>特定のレーン上のネットワークトラフィックに関する統計情報を取得します。</td>
<td>バケットが、システムのネットワークインタフェース上に構成されたネットワークレーンを含むときの受信および送信トラフィックに関する詳細情報を表示します。</td>
<td>433 ページの「レーン上のネットワークトラフィックに関する統計情報を取得する方法」</td>
</tr>
<tr>
<td>フロー上のトラフィックに関する統計情報を取得します。</td>
<td>ユーザーが定義したフローを含んでいる受信および送信トラフィックに関する情報を表示します。</td>
<td>436 ページの「フローに関する統計情報を取得する方法」</td>
</tr>
<tr>
<td>ネットワークトラフィックのアカウンティングを構成します。</td>
<td>アカウンティングの目的でトラフィック情報を取得するネットワークアカウンティングを構成します。</td>
<td>438 ページの「拡張ネットワークアカウンティングを構成する方法」</td>
</tr>
</tbody>
</table>

Oracle Solaris 管理：ネットワークインタフェースとネットワーク仮想化 • 2011年12月
リンク上のネットワークトラフィックに関する統計情報の収集

<table>
<thead>
<tr>
<th>タスク</th>
<th>説明</th>
<th>手順</th>
</tr>
</thead>
<tbody>
<tr>
<td>ネットワークトラフィックに関する履歴統計情報を取得します。</td>
<td>レーンやフローに関するネットワークトラフィックの履歴統計情報を取得するために、拡張ネットワークアカウンティングログファイルから情報を抽出します。</td>
<td>439ページの「ネットワークトラフィックに関する履歴統計情報を取得する方法」</td>
</tr>
</tbody>
</table>

フローを構成するための手順については、419ページの「フロー上のリソースの管理」を参照してください。これらの2つのコマンドについての詳細は、dlstat(1M)およびflowstat(1M)のマニュアルページを参照してください。

リンク上のネットワークトラフィックに関する統計情報の収集

dlstatコマンドとflowstatコマンドはそれぞれ、データリンクとフロー上のネットワークトラフィックに関する統計情報を監視および取得するためのツールです。これらのコマンドは、dladmコマンドとflowadmコマンドに対応しています。次の表は、*admコマンドのペアと*statコマンドのペアの間の対応、およびそれぞれの機能を示しています。

<table>
<thead>
<tr>
<th>管理コマンド</th>
<th>監視コマンド</th>
</tr>
</thead>
<tbody>
<tr>
<td>コマンド</td>
<td>コマンド</td>
</tr>
<tr>
<td>dladmコマンドオプション</td>
<td>データリンクを構成および管理するためのユーザーインタフェースとツール。</td>
</tr>
<tr>
<td>flowadmコマンドオプション</td>
<td>フローを構成および管理するためのユーザーインタフェースとツール。</td>
</tr>
</tbody>
</table>

dlstatコマンドの次のバリアントを使用すると、ネットワークトラフィック情報を収集できます。

- dlstat - システムによって送受信されているパケットに関する一般情報を表示します。
リンク上のネットワークトラフィックに関する統計情報の収集

- `dlstat show-phys` - 受信および送信リンクの使用に関する情報を表示します。このコマンドは、ネットワーク物理デバイスに関するトラフィック以外の情報を表示する `dladm show-phys` コマンドに対応しています。このコマンドが適用されるネットワークルーンのレベルの図については、図22-1を参照してください。

- `dlstat show-link` - 指定されたリンク上のトラフィックフローに関する詳細情報を表示します。このレーンは、そのデータリンクで識別されます。このコマンドは、データリンクに関するトラフィック以外の情報を表示する `dladm show-link` および `dladm show-vnic` コマンドに対応しています。`dlstat show-link` コマンドが適用されるネットワークルーンのレベルの図については、図22-1を参照してください。

- `dlstat show-aggr` - リンク集約でのポートの使用に関する情報を表示します。このコマンドは、リンク集約に関するトラフィック以外の情報を表示する `dladm show-aggr` コマンドに対応しています。

▼ ネットワークトラフィックに関する基本的な統計情報を取得する方法

1 管理者になります。
 詳細は、「Oracle Solaris の管理: セキュリティーサービス」の「管理権限を取得する方法」を参照してください。

2 すべてのデータリンク上の基本的なトラフィックフローを監視します。

```
# dlstat [-r|-t] [-i interval] [link]
```

- `-r` または `-t` は、受信側の統計情報のみ (`-r`オプション) または送信側の統計情報のみ (`-t`オプション) のどちらかを表示します。これらのオプションを使用しない場合は、受信側と送信側の両方の統計情報が表示されます。

- `-i interval` は、表示されている統計情報が更新される間隔(秒単位)を指定します。このオプションを使用しない場合は、静的な出力が表示されます。

- `link` は、指定されたデータリンクのみの統計情報を監視することを示します。このオプションを使用しない場合は、すべてのデータリンクに関する情報が表示されます。

単独で使用されると、`dlstat` コマンドは、構成されているすべてのデータリンク上の受信および送信パケットに関する情報を表示します。

次の情報は、`dlstat` コマンドで使用するほとんどどのオプションで表示されます。

- IP インタフェースが構成されていて、トラフィックを送受信できるシステム内のリンク
- パケットとパイトサイズ
- 割り込みと MAC ポーリングの統計情報
バケットチェーンの長さ

例 22-1 受信側と送信側の基本的な統計情報の表示
この例では、システム上の構成されているすべてのデータリンク上で受信されているネットワークトラフィックに関する情報を表示します。

```
# dlstat
LINK IPKTS RBYTES OPKTS OBYTES
e1000g0 101.88K 32.86M 40.16K 4.37M
nxge1 4.58M 6.78G 1.38M 90.90M
vnic1 8 336 0 0
```

例 22-2 受信側の統計情報の1秒間隔での表示
この例では、すべてのデータリンク上で受信されているトラフィックに関する情報を表示します。これらの情報は1秒ごとに更新されます。表示の更新を停止するためにCtrl-Cを押します。

```
# dlstat -r -i 1
LINK IPKTS RBYTES INTRS POLLS CH<10 CH10-50 CH>50
e1000g0 101.91K 32.86M 87.56K 14.35K 3.70K 205 5
nxge1 9.61M 14.47G 5.79M 3.82M 379.98K 85.66K 1.64K
vnic1 8 336 0 0 0 0 0
e1000g0 0 0 0 0 0 0 0
nxge1 82.13K 123.69M 50.00K 32.13K 3.17K 724 24
vnic1 0 0 0 0 0 0 0
```

```
^C
```

この出力では、割り込みの統計情報（INTRS）が重要です。割り込みの数が少ない場合は、パフォーマンスの効率が高いことを示します。割り込みの数が多い場合には、特定のリンクにさらに多くのリソースを追加することが必要になる場合があります。

例 22-3 送信側の統計情報の5秒間隔での表示
この例では、すべてのデータリンク上で送信されているトラフィックに関する情報を表示します。これらの情報は5秒ごとに更新されます。

```
# dlstat -t -i 5
LINK OPKTS OBYTES BLKCNT UBLCNT
e1000g0 40.24K 4.37M 0 0
nxge1 9.76M 644.14M 0 0
vnic1 0 0 0 0
e1000g0 0 0 0 0
nxge1 26.82K 1.77M 0 0
vnic1 0 0 0 0
```

```
^C
```
リンク使用状況に関する統計情報を取得する方法

1 管理者になります。詳細は、『Oracle Solaris の管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 リンクの統計情報を表示します。

```
# dlstat show-phys [-r|-t] [-i interval] [link]
```

- `[r|t]` 受信側の統計情報のみ (`-r`オプション) または送信側の統計情報のみ (`-t`オプション) のどちらかを表示します。これらのオプションを使用しない場合は、受信側と送信側の両方の統計情報が表示されます。
- `-i interval` 表示されている統計情報が更新される間隔(秒単位)を指定します。このオプションを使用しない場合は、静的な出力が表示されます。
- `link` 指定されたデータリンクのみの統計情報を監視することを示します。このオプションを使用しない場合は、すべてのデータリンクに関する情報が表示されます。

単独で使用されると、dlstat show-phys コマンドは、構成されているすべてのデータリンク上の受信および送信パケットに関する情報を表示します。

例 22-4 データリンクの受信リングの統計情報の表示

この例では、データリンクの受信リングの使用状況を表示します。

```
# dlstat show-phys -r nxge
```

```
LINK TYPE INDEX IPKTS RBYTES
nxge rx 0 21 1.79K
nxge rx 1 0 0
nxge rx 2 1.39M 2.10G
nxge rx 3 0 0
nxge rx 4 6.81M 10.26G
nxge rx 5 4.63M 6.97G
nxge rx 6 3.97M 5.98G
nxge rx 7 0 0
```

`nxge` デバイスには、INDEX フィールドで識別される 8 つの受信リングがあります。リンクごとのパケットの均一な分布は、それらのリンクが、各リンクの負荷に応じてリンクに正しく割り当てられていることを示す理想的な構成です。不均一な分布は、リンクごとのリンクの不均衡な分布を示していることがあります。この解決策は、NICによって、ユーザーがリンクごとにリンクを再分散できる動的なリング割り当てがサポートされているかどうかによって異なります。動的なリンク割り当てについての詳細は、399 ページの「送信リングと受信リング」を参照してください。
例 22-5 データリンクの送信リングの統計情報の表示

この例では、データリンクの送信リングの使用状況を表示します。

```
# dlstat show-phys -t nxge1

<table>
<thead>
<tr>
<th>LINK</th>
<th>TYPE</th>
<th>INDEX</th>
<th>OPKTS</th>
<th>OBYTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>nxge1 tx 0</td>
<td>44</td>
<td>3.96k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge1 tx 1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge1 tx 2</td>
<td>1.48M</td>
<td>121.68M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge1 tx 3</td>
<td>2.45M</td>
<td>201.11M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge1 tx 4</td>
<td>1.47M</td>
<td>120.82M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge1 tx 5</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge1 tx 6</td>
<td>1.97M</td>
<td>161.57M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge1 tx 7</td>
<td>4.59M</td>
<td>376.21M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge1 tx 8</td>
<td>2.43M</td>
<td>199.24M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge1 tx 9</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge1 tx 10</td>
<td>3.23M</td>
<td>264.69M</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

▼ レーン上のネットワークトラフィックに関する統計情報を取得する方法

1 管理者になります。

詳細は、『Oracle Solarisの管理: セキュリティーサービス』の「管理権限を取得する方法」を参照してください。

2 ネットワークレーンに関する統計情報を表示します。

```
# dlstat show-link [-r [F]|-t] [-i interval] [link]
```

- `-r` 送信側の統計情報のみ (`-r`オプション) または送信側の統計情報のみ (`-t`オプション) のどちらかを表示します。これらのオプションを使用しない場合は、受信側と送信側の両方の統計情報が表示されます。

- `-i interval` 表示されている統計情報が更新される間隔（秒単位）を指定します。このオプションを使用しない場合は、静的な出力が表示されます。

- `link` 指定されたデータリンクのみの統計情報を監視することを示します。このオプションを使用しない場合は、すべてのデータリンクに関する情報が表示されます。

リングのグループ化がサポートされており、専用リングが構成されている場合は、ハードウェアで対応する統計情報が表示されます。専用リングが構成されていない場合は、ソフトウェアで対応する統計情報が表示されます。
例22-6 レーンの受信側の統計情報の表示

この例では、次のような情報を表示します。

- パケットがハードウェアレーン上で受信されている状況
- パケットがソフトウェアレーン上で受信されている状況
- パケットがソフトウェアレーン上で受信されなかったと、割り当てられたCPUに
  ファンアウトされている状況

次のコマンドでは、特定のリンクの受信側の統計情報を表示します。これらの情報
は、リンク使用状況を示します。ただし、これらのデータには、帯域幅制限や優先
処理などのほかのリソース割り当ての実装も反映されている可能性があります。

```bash
# dlstat show-link -r nxge1
LINK TYPE ID INDEX IPKTS RBYTES INTRS POLLS CH<10 CH10-50 CH>50
nxge1 rx local -- 0 0 0 0 0 0 0
nxge1 rx hw 1 0 0 0 0 0 0
nxge1 rx hw 2 1.73M 2.61G 1.33M 400.22K 67.03K 7.49K 38
nxge1 rx hw 3 0 0 0 0 0 0 0
nxge1 rx hw 4 8.44M 12.71G 4.35M 4.09M 383.28K 91.24K 2.09K
nxge1 rx hw 5 5.68M 8.56G 3.72M 1.97M 203.68K 43.94K 854
nxge1 rx hw 6 4.90M 7.38G 3.11M 1.80M 168.59K 42.34K 620
nxge1 rx hw 7 0 0 0 0 0 0 0
```

次のコマンドでは、特定のリンクの受信側の統計情報を表示します。この出力
で、IDフィールドは、ハードウェアリングが推捨的に割り当てられているのか、ま
たはクライアントの間で共有されているのかを示します。ixgbe カードでは、リンク
上にVNICなどのほかのクライアントも構成されている場合はRxリングが共有され
ます。そのため、この特定の例では、IDフィールドのswの値で示されるようにRx
リングは共有されています。

```bash
# dlstat show-link -r ixgbe0
LINK TYPE ID INDEX IPKTS RBYTES INTRS POLLS CH<10 CH10-50 CH>50
ixgbe0 rx local -- 0 0 0 0 0 0 0
ixgbe0 rx local -- 794.28K 1.19G 794.28K 0 0 0 0 0
```

次のコマンドは、特定のリンクの受信側の統計情報の使用法を示しています。さら
に、このコマンドでは-Fオプションが使用されているため、出力にはファンアウト
情報も表示されます。具体的には、ファンアウト数は2つ(0と1)です。リング0を
使用するハードウェアリング上で受信されたネットワークトラフィックは分割さ
れ、2つのファンアウトにわたって渡されます。同様に、リング1を使用するハード
ウェアリング上で受信されたネットワークトラフィックも分割され、2つのファンア
ウトにわたって分配されます。

```bash
# dlstat show-link -r -F nxge1
LINK ID INDEX FOUT IPKTS
nxge1 local -- 0 0
nxge1 hw 0 0 382.47K
nxge1 hw 0 1 0
```

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
例 22-7 レーンの送信側の統計情報の表示

次の例では、特定のレーン上の送信パケットに関する統計情報を表示します。

```
# dlstat show-link -t nxge1
```

<table>
<thead>
<tr>
<th>LINK</th>
<th>TYPE</th>
<th>ID</th>
<th>INDEX</th>
<th>OPKTS</th>
<th>OBYTES</th>
<th>BLKCNT</th>
<th>UBLKCNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>nxge1</td>
<td>hw</td>
<td>1</td>
<td>0</td>
<td>367.50K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nxge1</td>
<td>hw</td>
<td>1</td>
<td>1</td>
<td>433.24K</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

フロー上のネットワークトラフィックに関する統計情報の収集

フローの統計情報は、システム上で定義されている任意のフロー上のパケットトラフィックを評価するのに役立ちます。フロー情報の取得するには、flowstat コマンドを使用します。このコマンドについての詳細は、flowstat(1M) のマニュアルページを参照してください。

flowstat コマンドのもっとも一般的に使用される構文は次のとおりです。

```
# flowstat [-r|-t] [-i interval] [-l link flow]
```

- **[-r|-t]** 受信側の統計情報のみ(-rオプション) または送信側の統計情報のみ(-tオプション) のどちらかを表示します。これらのオプションを使用しない場合は、受信側と送信側の両方の統計情報が表示されます。

- **[-i interval]** 表示されている統計情報が更新される間隔 (秒単位) を指定します。このオプションを使用しない場合は、静的な出力が表示されます。

- **link** 指定されたデータリンク上のあるフローの統計情報を監視する内容を示します。このオプションを使用しない場合は、すべてのデータリンク上のあるフローに関する情報が表示されます。

- **flow** 指定されたフローのみの統計情報を監視することを示します。このオプションを使用しない場合は、リンクを指定したかどうかに応じて、すべてのフローの統計情報が表示されます。
フローに関する統計情報を取得する方法

始める前に flowstat コマンドは、ネットワーク構成内にフローが存在する場合にのみ使用できます。フローを構成するには、第21章「ネットワーククリソースの管理」を参照してください。

1 以前にフロー制御を構成したシステムで、大域ゾーンの管理者になります。 詳細は、『Oracle Solaris の管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 フロー上のネットワークトラフィックを監視する方法のシミュルを取得するには、次のコマンドのいずれかを実行します。
   - すべてのフロー上の受信および送信パケットに関する統計情報を表示します。
     
     ```
     # flowstat
     ```
     このコマンドでは、構成されているすべてのフロー上のトラフィック情報の静的な表示が提供されます。
   - すべてのフローに関する基本的なネットワークトラフィック統計情報を指定された間隔で表示します。
     
     ```
     # flowstat -i interval
     ```
     この統計情報の表示は、ユーザーがCtrl-Cを押して出力の生成を停止するまで、指定された間隔で更新されます。
   - 指定されたデータリンク上に構成されているすべてのフロー上の受信パケットに関する統計情報を表示します。
     
     ```
     # flowstat -r -l link
     ```
   - 指定されたフロー上の送信パケットに関する統計情報を指定された間隔で表示します。
     
     ```
     # flowstat -t -i interval flow
     ```

例 22-8 すべてのフローのトラフィック統計情報の1秒間隔での表示

この例では、システム上で構成されているすべてのフロー上の受信および送信トラフィックに関する情報を1秒ごとに表示します。

```
# flowstat -i 1
FLOW IPKTS RBYTES IERRS OPKTS OBYTES OERRS
flow1 528.45K 787.39M 0 179.39K 11.85M 0
flow2 742.81K 1.10G 0 0 0 0
flow3 0 0 0 0 0 0
flow1 67.73K 101.02M 0 21.04K 1.39M 0
flow2 0 0 0 0 0 0
flow3 0 0 0 0 0 0
```

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月
例 22-9 すべてのフローの送信側の統計情報の表示

```
# flowstat -t
FLOW  OPKTS   OBYTES  OERRS
flow1 24.37M  1.61G  0
flow2  0      0      0
flow1  4      216   0
```

例 22-10 指定されたリンク上のすべてのフローの受信側の統計情報の表示

この例では、データリンク net0 上に作成されたすべてのフローのハードウェアレーン内の受信トラフィックを表示します。

```
# flowstat -r -i 2 -l net0
FLOW  IPKTS   RBYTES  IERRS
tcp-flow 183.11K 270.24M  0
udp-flow  0      0      0
tcp-flow  373.83K 551.52M  0
udp-flow  0      0      0
tcp-flow  372.35K 549.04M  0
udp-flow  0      0      0
tcp-flow  372.87K 549.61M  0
udp-flow  0      0      0
tcp-flow  371.57K 547.89M  0
udp-flow  0      0      0
tcp-flow  191.92K 282.95M  0
udp-flow  206.51K 310.70M  0
tcp-flow  0      0      0
udp-flow  222.75K 335.15M  0
tcp-flow  0      0      0
udp-flow  223.00K 335.52M  0
tcp-flow  0      0      0
udp-flow  160.22K 241.07M  0
tcp-flow  0      0      0
udp-flow  167.89K 252.61M  0
tcp-flow  0      0      0
udp-flow  9.52K  14.32M  0
```

ネットワークアカウンティングの設定

ネットワークアカウンティング機能を使用すると、ネットワークトラフィックに関する統計情報をログファイル内に取得できます。この方法では、追跡、プロビジョニング、統合、および請求の目的でトラフィックのレコードを保持することができ、あとで、一定期間にわたるネットワーク使用に関する履歴情報を取得するために、このログファイルを参照できます。
ネットワークアカウンティングの設定

拡張アカウンティング機能を構成するには、acctadmコマンドを使用します。

▼ 拡張ネットワークアカウンティングを構成する方法

1. ネットワーク使用を追跡するインタフェースを備えたシステムで、管理者にします。
   詳細は、「Oracle Solarisの管理: セキュリティサービス」の「管理権限を取得する方法」を参照してください。

2. システム内の拡張ネットワークアカウンティングのステータスを表示します。
   
   # acctadm net

   acctadmコマンドを使用して、次の4つのタイプの拡張アカウンティングを有効にすることができます。
   
   - プロセスアカウンティング
   - タスクアカウンティング
   - IPQoS(IP Quality of Service)のためのフローアカウンティング
   - リンクとフローのためのネットワークアカウンティング

   netを指定すると、ネットワークアカウンティングのステータスが表示されますが、netが使用されない場合は、4つのすべてのアカウンティングタイプのステータスが表示されます。

   注 - ネットワークアカウンティングはまた、419ページの「フロー上のリソースの管理」で説明されているflowadmおよびflowstatコマンドで管理されるフローにも適用されます。そのため、これらのフローのためのアカウンティングを設定するには、acctadmコマンドでnetオプションを使用します。フローアカウンティングを有効にし、IPQoS構成に適用されるflowオプションは使用しないでください。

3. ネットワークトラフィックのための拡張アカウンティングを有効にします。
   
   # acctadm -e extended -f filename net

   ここで、filenameには、ネットワークトラフィックの統計情報を取得するログファイルのフルパスが含まれます。このログファイルは、指定した任意のディレクトリ内に作成できます。

4. 拡張ネットワークアカウンティングがアクティブになっていることを確認します。
   
   # acctadm net

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化 • 2011年12月
ネットワークトラフィックのための拡張アカウンティングの構成

この例では、データリンクおよびシステム上で構成されている任意のフロー上のネットワークトラフィックに関する履歴情報を取得して表示する方法を示しています。

最初に、次のように、すべてのアカウンティングタイプのステータスを表示します。

```
# acctadm
Task accounting: inactive
Task accounting file: none
Tracked task resources: none
Untracked task resources: extended
Process accounting: inactive
Process accounting file: none
Tracked process resources: none
Untracked process resources: extended, host
Flow accounting: inactive
Flow accounting file: none
Tracked flow resources: none
Untracked flow resources: extended
Network accounting: inactive
Network accounting file: none
Tracked Network resources: none
Untracked Network resources: extended
```

この出力は、ネットワークアカウンティングがアクティブになっていないことを示します。

次に、拡張ネットワークアカウンティングを有効にします。

```
# acctadm -e extended -f /var/log/net.log net
# acctadm net
Net accounting: active
Net accounting file: /var/log/net.log
Tracked net resources: extended
Untracked net resources: none
```

ネットワークアカウンティングを有効にしたら、dlstatおよびflowstatコマンドを使用して、ログファイルから情報を抽出できます。次の手順は、このための手順について説明しています。

▼ ネットワークトラフィックに関する履歴統計情報取得する方法

始める前に

ネットワークに関する履歴データを表示するには、その前にネットワークのための拡張アカウンティングを有効にする必要があります。さらに、フロー上のトラ
ネットワークアカウンティングの設定

フィックに関する履歴データを表示するには、まず 419 ページの「フロー上のリソースの管理」の説明に従ってシステム内のフローを構成する必要があります。

1 ネットワーク使用を追跡するインタフェースを備えたシステムで、管理者になります。
詳細は、『Oracle Solaris の管理: セキュリティサービス』の「管理権限を取得する方法」を参照してください。

2 データリンク上のリソース使用状況に関する履歴情報抽出して表示するには、次のコマンドを使用します。

```
# dlstat show-link -h [-a] -f filename [-d date] [-F format] [-s start-time] [-e end-time] [link]
```
- `-h` リソース使用状況に関する履歴情報の概要をデータリンク上の受信および送信バケットごとに表示します。
- `-a` すべてのデータリンク (データ取得のあとにすでに削除されたデータリンクを含む) 上のリソース使用状況を表示します。
- `-f filename` acctadm コマンドでネットワークアカウンティングが有効にされたときに定義されたログファイルを指定します。
- `-d` 情報が使用可能な日付のログ記録された情報を表示します。
- `-F format` データを特定の形式で表示します。現在、サポートされている形式は gnuplot だけです。
- `-s start-time,` 指定された日付と時刻の範囲の使用可能なログ記録された情報を表示します。MM/DD/YYYY,hh:mm:ss の形式を使用します。hour (hh) には、24 時間制のクロック表記を使用する必要があります。日付を含めない場合は、現在の日付の時間範囲のデータが表示されます。
- `-e end-time` 指定されたデータリンクの履歴データを表示します。このオプションを使用しない場合は、構成されているすべてのデータリンクの履歴ネットワークデータが表示されます。

3 構成されているフロー上のネットワークトラフィックに関する履歴情報を抽出して表示するには、次のコマンドを使用します。

```
# flowstat -h [-a] -f filename [-d date] [-F format] [-s start-time] [-e end-time] [flow]
```
- `-h` リソース使用状況に関する履歴情報の概要をデータリンク上の受信および送信バケットごとに表示します。
- `-a` すべてのデータリンク (データ取得のあとにすでに削除されたデータリンクを含む) 上のリソース使用状況を表示します。
- `-f filename` acctadm コマンドでネットワークアカウンティングが有効にされたときに定義されたログファイルを指定します。
-d  情報が使用可能な日付のログ記録された情報を表示します。
-F format  データを特定の形式で表示します。現在、サポートされている形式はgnuplotだけです。
-s start-time,  指定された日付と時刻の範囲の使用可能なログ記録された情報を表示します。MM/DD/YYYY, hh:mm:ssの形式を使用します。hour(hh)には、24時間制のクロック表記を使用する必要があります。日付を含める場合は、現在の日付の時間範囲のデータが表示されます。
-e end-time  指定されたデータリンクの履歴データを表示します。このオプションを使用しない場合は、構成されているすべてのデータリンクの履歴ネットワークデータが表示されます。

例22-12 データリンク上のリソース使用状況に関する履歴情報の表示
次の例は、指定されたデータリンク上のネットワークトラフィックとそのリソースの使用に関する履歴情報メッセージを示しています。

```
# dlnstat show-link -h -f /var/log/net.log
```

<table>
<thead>
<tr>
<th>LINK</th>
<th>DURATION</th>
<th>IPACKETS</th>
<th>RBYTES</th>
<th>OPACKETS</th>
<th>OBYTES</th>
<th>BANDWIDTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1000g0</td>
<td>80</td>
<td>1031</td>
<td>546908</td>
<td>0</td>
<td>0</td>
<td>2.44 Kbps</td>
</tr>
</tbody>
</table>

例22-13 フロー上のリソース使用状況に関する履歴情報の表示
次の例は、フロー上のネットワークトラフィックとそのリソースの使用に関する履歴統計情報を表示するための各種の方法を示しています。

リソース使用状況の履歴統計情報をフロー上のトラフィックごとに表示します。

```
# flowstat -h -f /var/log/net.log
```

```
FLOW DURATION IPACKETS RBYTES OPACKETS OBYTES BANDWIDTH
flowtcp 100 1031 546908 0 0 43.76Kbps
flowudp 0 0 0 0 0 0.00Mbps
```

指定された日付と時刻の範囲にわたるリソース使用状況の履歴統計情報をフロー上のトラフィックごとに表示します。

```
# flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 -f /var/log/net.log flowtcp
```

<table>
<thead>
<tr>
<th>FLOW</th>
<th>START</th>
<th>END</th>
<th>RBYTES</th>
<th>OBYTES</th>
<th>BANDWIDTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowtcp</td>
<td>10:39:06</td>
<td>10:39:26</td>
<td>1546</td>
<td>6539</td>
<td>3.23 Kbps</td>
</tr>
<tr>
<td>flowtcp</td>
<td>10:39:26</td>
<td>10:39:46</td>
<td>3586</td>
<td>9922</td>
<td>5.40 Kbps</td>
</tr>
</tbody>
</table>

第22章・ネットワークトラフィックとリソース使用状況の監視 441
ネットワークアカウンティングの設定

<table>
<thead>
<tr>
<th>flowtcp</th>
<th>10:39:46</th>
<th>10:40:06</th>
<th>240</th>
<th>216</th>
<th>182.40 bps</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowtcp</td>
<td>10:40:06</td>
<td>10:40:26</td>
<td>0</td>
<td>0</td>
<td>0.00 bps</td>
</tr>
</tbody>
</table>

指定された日付と時刻の範囲にわたるリソース使用状況の履歴統計情報をフロー上のトラフィックごとに表示します。これらの情報はgnuplot形式を使用して表示されます。

```
# flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-F gnuplot -f /var/log/net.log flowtcp
# Time tcp-flow
10:39:06 3.23
10:39:26 5.40
10:39:46 0.18
10:40:06 0.00
```
用語集

3DES  Triple-DES を参照してください。

AES  Advanced Encryption Standard の略。対称128 ビットブロックのデータ暗号技術。2000 年
の10月、米国政府は暗号化標準としてこのアルゴリズムのRijndael方式を採用しました。AESはDES
に代わる米国政府の標準として採用されています。

Blowfish  32 ビットから 448 ビットまでの可変長キーの対称ブロックの暗号化アルゴリズム。その
作成者であるBruce Schneier氏は、鍵を頻繁に変更しないアプリケーションに効果的であると述べています。

CA  認証局 (CA) を参照してください。

DEPRECATED アドレス  IPMP グループ内でデータの発信元アドレスとして使用することのできないIP アドレ
ス。通常、IPMP の検査用 IP アドレスはDEPRECATED アドレスです。ただし、任意のアドレ
スにDEPRECATED のマークを付けて、そのアドレスが発信元アドレスとして使用される
ことを防止できます。

DES  Data Encryption Standard、1975年に開発され、1981年にANSI X.3.92としてANSIで標準化された対称鍵の暗号化方式。DESでは56 ビットの鍵を使用します。

Diffie-Hellman プロトコル  公開鍵暗号化としても知られています。1976年にDiffie氏とHellman氏が開発した非対称暗号鍵協定プロトコルです。このプロトコルを使用すると、セキュリティ保護されていない伝達手段で、事前の秘密情報がなくても 2 人のユーザーが秘密鍵を交換できます。Diffie-Hellman は、IKE プロトコルで使用されます。

diffserv モデル  IP ネットワークで差別化サービスを実装するためのIETF (Internet Engineering Task Force)
のアーキテクチャ標準。主なモジュールとして、クラスファイ
ア、メーター、マーカー、スケジューラ、およびドロッパがあります。IPoQoS では、ク
ラスファイア、メーター、およびマーカーの各モジュールを実装します。diffserv モデル
については、RFC 2475 (An Architecture for Differentiated Services) に解説されています。

DSA  デジタル署名アルゴリズム。512 ビットから 4096 ビットまでの可変長キーの公開鍵アル
ゴリズム。米国政府標準である DSS は最大 1024 ビットです。この場合、DSA では入力に
SHA-1 を使用します。

DS コードポイント (DSCP)  IP ヘッダーのDSフィールドに含まれていて、パケットの転送方法を指示する6 ビット
の値。

header  IP ヘッダーを参照してください。
HMAC
メッセージ認証を行うためのキー付きハッシュ方法。HMAC は秘密鍵認証アルゴリズムの一つです。HMAC は秘密共有鍵と併用して、MD5、SHA-1 などの繰り返し暗号化のハッシュ関数で使用します。HMAC の暗号の強さは、基となるハッシュ関数のプロパティによって異なります。

ICMP
インターネット制御メッセージプロトコル (Internet Control Message Protocol)。エラーの処理や制御メッセージの交換に使用されます。

ICMP エコー要求パケット
応答を促すためにインターネット上のマシンに送信されるメッセージ。そのようなパケットは一般に “ping” パケットといわれています。

IKE
インターネットキー交換。IPsec セキュリティーアーキテクチャ (SA) 用の認証された鍵情報の供給を自動化します。

IP
インターネットプロトコル (IP)、IPv4、IPv6 を参照してください。

IPMP グループ
IP マルチバスグループ。ネットワークの可用性と利用率を向上させるために相互に入れ替え可能なものとしてシステムで扱われる。一連のネットワークインタフェースと一連のデータアドレスで構成されます。IPMP グループは、そのすべての IP インタフェースとデータアドレスを含めて、IPMP インタフェースによって表されます。

IPQoS
diffserv 標準に加えて、仮想 LAN に対するフローアグリプレイングや 802.1D マーカーの実装を行うソフトウェア機能。IPQoS を使用すると、IPQoS 構成ファイル内に定義したとおりに、さまざまなレベルのネットワークサービスを顧客やアプリケーションに提供できます。

IPsec
IP セキュリティ。IP データグラムを保護するためのセキュリティーアーキテクチャ。

IPv4
インターネットプロトコルのバージョン 4。単に IP と呼ばれることもあります。このバージョンは 32 ビットのアドレス空間をサポートしています。

IPv6
インターネットプロトコルのバージョン 6。128 ビットのアドレス空間をサポートしています。

IP スタック
TCP/IP はしばしば「スタック」と呼ばれますが。データ交換のクライアントエンドとサーバーイエンドですべてのデータが通過する層 (TCP、IP、場合によってはそのほかを含む) のことを意味します。

IP データグラム
IP 経由で転送される情報パケット。IP データグラムはヘッダーとデータを含みます。ヘッダーにはデータグラムのソースと宛先のアドレスが含まれます。ヘッダーのその他のフィールドには、複数のデータグラムを宛先で識別し、再結合するための情報が含まれます。

IP 内 IP カプセル化
IP パケット内で IP パケットをトンネリングするためのメカニズム。

IP ヘッダー
インターネットパケットを固有に識別する 20 バイトのデータ。ヘッダーには、パケットの発信元と着信先のアドレスが含まれています。さらに、ヘッダー内のオプションによって、新しいパケットを追加できます。
IP リンク
リンク層でノード間の通信に使用される通信設備や通信メディア。リンク層とはIPv4およびIPv6のすぐ下の層です。例としては、Ethernet (ブリッジされたものも含む) や ATMネットワークなどがあります。IPv4のサブネット番号またはネットワーク接頭辞がIPリンクに割り当てられます。同じサブネット番号またはネットワーク接頭辞を複数のIPリンクに割り当てることができません。ATM LANEでは、1つのエミュレートされたLANで、ARPを使用する場合、ARPプロトコルの有効範囲は単一のIPリンクです。

MD5
デジタル署名などのメッセージ認証に使用する繰り返し暗号化のハッシュ関数。1991年にRivest氏によって開発されました。

MTU
最大転送単位。リンクに転送できるサイズ (オクテット単位)。たとえば、EthernetのMTUは1500オクテットです。

NAT
ネットワークアドレス変換を参照してください。

Perfect Forward Secrecy (PFS)
PFSでは、データ伝送を保護するために使用されるキーが、追加のキーを導き出すために使用されることはありません。さらに、データ伝送を保護するために使用されるキーのソースが、追加のキーを導き出すために使用されることはありません。

PKI
Public Key Infrastructure。インターネットトラザクションに関与する各関係者の有効性を確認および承認する、デジタル署名、認証局、ほかの登録機関のシステム。

RSA
デジタル署名と公開鍵暗号化システムを提供するための方法。その発明者であるRivest氏、Shamir氏、Adleman氏によって1978年に最初に公開されました。

SA
セキュリティーアーキシエンション (SA) を参照してください。

SADB
セキュリティーアーキシエンションデータベース。暗号化鍵と暗号化アルゴリズムを指定するテーブル。鍵とアルゴリズムは、安全なデータ転送で使用されます。

SCTP
「ストリーム制御転送プロトコル」を参照してください。

SHA-1
セキュリティー保護されたハッシュアルゴリズム。メッセージ暗号を作成するために2^64文字以下を入力するときに使用します。SHA-1アルゴリズムはDSAに入力されます。

smurf攻撃
リモートアドレスからIPブロードキャストアドレスまたは複数のブロードキャストアドレスに向けられたICMP echo requestパケットを使用して、深刻なネットワークの摩擦や中断を引き起こすこと。

SPD
セキュリティーポリシーデータベース (SPD) を参照してください。

SPI
セキュリティーパラメータインデックス (SPI) を参照してください。

TCP/IP
TCP/IP (通信制御プロトコル/インターネットプロトコル) は、インターネットの基本的な通信言語またはプロトコルです。ブリベートネットワーク (イントラネットやエクストラネット) の通信プロトコルとしても使用されます。
Triple-DES

Triple-DES は、三重データ暗号化規準の1つ。3DES は「3DES」と略記することもあります。

インターシャプログラカル

インターネットを介してデータのあるコンピュータから別のコンピュータに送信するための方法またはプロトコル。

エニーキャストアドレス

一般的に別のノードに属するインタフェースグループに割り当てられる IPv6 アドレス。エニーキャストアドレスに送られたパケットは、そのアドレスを持つ、プロトコルに基づき「最も近い」インタフェースに配信されます。パケットの経路制御は、経路制御プロトコルの距離測定に応じて決定されます。

エニーキャストグループ

同じエニーキャスト IPv6 アドレスからなるインタフェースグループ。IPv6 の Oracle Solaris 実装は、エニーキャストアドレスやグループの作成をサポートしていません。ただし、Oracle Solaris IPv6 ノードはトラフィックをエニーキャストグループに送信できます。

解釈ドメイン

データ形式や、ネットワークトラフィック交換タイプ、セキュリティー関連情報の命名規約を定義します。セキュリティー関連情報の例としては、セキュリティポリシーや、暗号化アルゴリズム、暗号化モードなどが含まれます。

回復検出

障害の発生後、NIC や NIC からレイヤー3デバイスへの経路が、正しく動作し始めたことを検出する処理。

鍵管理

セキュリティーアーザーシェーション (SA) を管理する方法。

仮想 LAN

仮想ネットワーク

仮想ネットワークインタフェース

仮想プライベートネットワーク

カプセル化

カプセル化セキュリティーベイロード

データウェアおよびハードウェアのネットワークリソースとネットワーク機能を組み合わせたもの。単一のソフトウェアエンティティーとしてまとめて管理されます。「内部」仮想ネットワークは、ネットワークリソースを単一のシステムに統合したものです。「ネットワーク(-network in a box)」と呼ばれることもあります。

物理的なネットワークインタフェースで構成されているかどうかに関係なく、仮想ネットワーク接続を提供する擬似インタフェース。他の IP ソーンや xVM メインなどのコンテナが VNIC 上に構成されて、仮想ネットワークを形成します。

インターネットのような公共ネットワーク内でトンネルを利用する、単独の、安全で論理的なネットワーク。

ヘッダーとベイロードを1 番目のパケット内に配置し、そのパケットを2 番目のパケットのベイロード内に配置すること。

データグラムに対して認証と完全性を提供する拡張ヘッダー。ESP は、IP セキュリティーアーキテクチャー (IPsec) の5つのコンポーネントの1つです。
キーストア名
管理者がストレージ領域(つまり、キーストア)に与える、ネットワークインタフェースカード(NIC)上の名前。キーストア名は、「トークン」、「トークンID」とも呼ばれます。

逆方向トンネル
モバイルノードの気付アドレスで始まり、ホームエージェントで終わるトンネル。

近傍検索
接続されているリンク上にあるほかのホストをホストが特定できるようにするためのIPメカニズム。

近傍通知
近傍要請メッセージに対する応答、またはデータリンク層アドレスの変更を通知するために、ノードが自発的に近傍通知メッセージを送ること。

近傍要請
近傍のリンク層アドレスを決定するために、ノードによって送信される要請。また、キャッシュされたリンク層アドレスによって近傍が到達可能であるかを確認します。

クラス
IPQoSでは、似たような特性を共有するネットワークフローのグループ。クラスは、IPQoS構成ファイル内に定義します。

クラスレスドメイン間経路制御
ネットワーククラス(クラスA、B、C)に基づかないIPv4アドレス形式。CIDRアドレスの長さは32ビットです。標準的なIPv410進ドット表記形式にネットワーク接頭辞を付加したものを使用します。この接頭辞はネットワーク番号とネットワークマスクを定義します。

結果
トラフィックの計測結果に基づいて実行されるアクション。IPQoSメーターには、赤、黄、および緑の3種類の結果(outcome)があり、IPQoS構成ファイル内に定義されます。

検査用アドレス
IPMPグループ内のIPアドレスで、検査信号用の発信元アドレスまたは宛先アドレスとして使用する必要があり、データトラフィック用の発信元アドレスまたは宛先アドレスとして使用してはならないもの。

公開鍵暗号化
2つの鍵を使用する暗号化システム。公開鍵はだれでも知ることができます。非公開鍵は、メッセージの受信者だけが知っています。IKEにより、IPsecの公開鍵が提供されます。

再実行攻撃
IPsecでは、パケットが侵入者によって捕获されるような攻撃のこと。格納されたパケットは、あとで元のパケットを置き換えるか繰り返します。そのような攻撃を防止するために、パケットを保護している秘密鍵が存在している間、値が増加する続けるフィールドをパケットに含めることができます。

最小カプセル化
ホームエージェント、外来エージェント、およびモバイルノードによってサポートされる任意の形態のIPv4内IPv4トンネリング。最小カプセル化は、IP内IPカプセル化よりも8ないし12バイト少ないオーバヘッドしか持ちません。

サイトローカルアドレス
単一サイト上でアドレスを指定するために使用します。

自動設定
ホストが、サイト接頭辞とローカルMACアドレスからそのIPv6アドレスを自動的に設定する処理。
<table>
<thead>
<tr>
<th>自動検出</th>
<th>CAが無効とされた公開鍵証明書のリスト。CRLは、IKEを使用して管理されるCRLデータベースに格納されます。</th>
</tr>
</thead>
<tbody>
<tr>
<td>スタック</td>
<td>IPスタックを参照してください。</td>
</tr>
<tr>
<td>ステートフルパケットフィルタ</td>
<td>アクティブな接続の状態を監視し、そこから得た情報を使ってパケットフィルタを通じさせるネットワークパケットを検出するフィルター。要求と応答を追跡、照合することによって、ステートフルパケットフィルタは、要求と一致しない応答を選別できます。</td>
</tr>
<tr>
<td>ステートレス自動設定</td>
<td>ホストがそれ自身のIPv6アドレスを生成する処理。その生成は、ホスト自身のMACアドレスとルーチンIPv6ルーターによって生成されるIPv6接頭辞を結合することによって行われます。</td>
</tr>
<tr>
<td>ストリーム制御転送プロトコル</td>
<td>TCPと似た方法で接続指向の通信を行う転送層プロトコル。さらに、このプロトコルは、接続のエンドポイントの1つが複数のIPアドレスをもつことができる複数ホーム機能をサポートします。</td>
</tr>
<tr>
<td>スプーフィング</td>
<td>コンピュータに位置を保つように、メッセージが、信頼されるホストから来たかのように見えるIPアドレスを使ってコンピュータにメッセージを送信すること。IPのなりすましを行うために、ハッカーは、さまざまなテクニックを使って、信頼されるホストのIPアドレスを見つけ、次にパケットヘッダーを変更します。それによって、パケットは、そのホストから来たかのように見えます。</td>
</tr>
<tr>
<td>セキュリティーアソシエーション (SA)</td>
<td>1つのホストから2つのホストにセキュリティー属性を指定するアソシエーション。</td>
</tr>
<tr>
<td>セキュリティーバラメータインデックス (SPI)</td>
<td>受信したパケットを復号化するために使用する、SADB (セキュリティーアソシエーションデータベース) 内の行を特定する整数値。</td>
</tr>
<tr>
<td>セキュリティーポリシーデータベース (SPD)</td>
<td>パケットにどのレベルの保護を適用するかを指定するデータベース。SPDは、IPトラフィックのフィルタとして、パケットを破棄すべきか、検証済みとして通過させるべきか、IPsecで保護すべきかを決めます。</td>
</tr>
<tr>
<td>セレクタ</td>
<td>ネットワークトシームからトラフィックを選択するために、特定クラスのパケットに適用される条件を具体的に定義する要素。セレクタは、IPQoS構成ファイルのフィルタ句に定義します。</td>
</tr>
<tr>
<td>専用アドレス</td>
<td>インターネット経由で経路制御できないIPアドレス。プライベートアドレスは、インターネット接続を必要としない社内ネットワークのホストで使用できます。このようなアドレスはAddress Allocation for Private Internets (<a href="http://www.ietf.org/rfc/rfc1918.txt?number=1918)%E3%81%A7%E5%AE%9A%E7%BE%A9%E3%81%95%E3%82%8C%E3%80%81%E3%81%97%E3%81%B0%E3%81%97%E3%81%B0%E2%80%9C1918%E2%80%9D%E3%82%A2%E3%83%89%E3%83%AC%E3%82%B9%E3%81%A8%E5%91%BC%E3%81%B0%E3%82%8C%E3%81%A6%E3%81%84%E3%81%BE%E3%81%99%E3%80%82">http://www.ietf.org/rfc/rfc1918.txt?number=1918)で定義され、しばしば“1918”アドレスと呼ばれています。</a></td>
</tr>
<tr>
<td>双方向トンネル</td>
<td>双方向にデータグラムを送信するトンネル。</td>
</tr>
</tbody>
</table>
ネットワークアドレス変換
NAT。あるネットワークで使用されているIPアドレスを、別のネットワークで認識されている異なるIPアドレスに変換すること。必要となる大域IPアドレスの数を抑えるために使用されます。

ネットワークインタフェースカード (NIC)
ネットワークへのインタフェースになる、ネットワークアダプタカード。NICによっては、1Gbカードなど複数の物理インタフェースを装備できるものもあります。

ノード
IPv6では、IPv6が有効なシステムのこと。ホストかルーターかは問いません。
パケット | 通信回線上で、1単位として送られる情報の集合。IPヘッダーやバイロードを含みます。
---|---
パケットフィルタ | 指定するパケットのファイアウォールの通過を許可するようにも許可しないようにも設定できるファイアウォール機能。
パケットヘッダー | IPヘッダーを参照してください。
ハッシュ値 | テキストの文字列から生成される数値。ハッシュ値数は、転送されるメッセージが改ざんされないようにするために使用します。一方向のハッシュ値の例としては、MD5とSHA-1があります。
非対称鍵暗号化 | メッセージの送信側で異なる鍵を使用してメッセージの暗号化および暗号解除を行う暗号化システム。非対称鍵を使用して、対称暗号に対するセキュリティ保護されたチャネルを作成します。Diffie-Hellmanプロトコルは、非対称鍵プロトコルの例です。対称暗号化と比較してください。
ファイアウォール | 組織のプライベートネットワークやインターネットをインターネットから切り離し、外部からの侵入を防ぐためのデバイスまたはソフトウェア。ファイアウォールには、フィルタリングや、プロキシサーバー、NAT(ネットワークアドレス変換)などを組み込むことができます。
フィルタ | クラスの特性をIPQoS構築ファイル内で定義するための規則セット。IPQoSシステムでは、IPQoS構築ファイル内で定義されたフィルタに適合するトラフィックフローを選択して処理します。パケットフィルタを参照してください。
負荷分散 | インターネットまたはアウトバウンドのトラフィックを一連のインタフェースに分散する処理。負荷分散を適用するとき、より高いスループットを達成できます。ただし、負荷分散が行われるのは、データが複数の接続を経由して複数の標識に送信される場合だけです。負荷分散には、インターコードトラフィック用のインターコード負荷分散アンドトラフィック用のアウトバウンド負荷分散の2種類があります。
物理インタフェース | リンクへのシステムの接続。この接続は通常、デバイスドライバとネットワークインタフェースカード(NIC)として実装されます。NICによっては、igbのように複数の接続点を持つものもあります。
フローアウンダリング | IPQoSでは、トラフィックフローに関する情報を蓄積、記録する処理のこと。フローアウンダリングを確立するには、flowacctモジュールのパラメータをIPQoS構築ファイル内で定義します。
ブロードキャストアドレス | アドレスのホスト部分のビットがすべてゼロ(10.50.0.0)か1(10.50.255.255)であるIPv4ネットワークアドレス。ローカルネットワーク上のマシンからブロードキャストアドレスに送信されたパケットは、同じネットワーク上のすべてのマシンに配信されます。
プロキシサーバー | Webブラウザなどのクライアントアプリケーションと別のサーバーの間にあるサーバー。要求をフィルタするために使用されます(たとえば、特定のWebサイトへのアクセスを防ぐ)。
プロトコルスタック | IPスタックを参照してください。
リダイレクト

ペイロード バケットで伝送されるデータ。ペイロードには、バケットを宛先に送るために必要なヘッダー情報は含まれません。

ホスト バケット転送を行わないシステム。Oracle Solaris をインストールされると、システムはデフォルトでホストになります。つまり、このシステムはバケットを転送できません。通常、ホストは 1 つの物理インタフェースをもつ。ただし、複数のインタフェースをもつこともできます。

ホップ 2 つのホストを分離するルーターの数を判別するための手段。たとえば、始点ホストと終点ホストが 3 つのルーターで分離されている場合、ホストは互いに 4 ホップ離れています。

ホップ単位動作 (Per-Hop Behavior、 PHB) トランジククラスに割り当てられる優先順位。PHB は、そのクラスのフローに割り当てられる、ほかのトランジククラスに対する相対的な優先度を示します。

マーカー 1. diffserv アーキテクチャおよびIPQoS のモジュールの 1つ。バケットの転送方法を指示する値を IP パケットの DS フィールドに付けます。IPQoS 実装では、このマーカーモジュールは dscp mk です。

2. IPQoS 実装のモジュールの 1つ。ユーザー優先順位の値を Ethernet データグラムの仮想 L AN タグに付けます。ユーザー優先順位の値は、VLAN デバイスを備えたネットワーク上でデータグラムが転送される方法を示します。このモジュールは dtrace mk と呼ばれています。

マルチキャストアドレス 特定の方法でインタフェースのグループを特定する IPv6 アドレス。マルチキャストアドレスは、グループにあるすべてのインタフェースに配信されます。IPv6 マルチキャストアドレスには、IPv4 ブロードキャストアドレスに似た機能があります。

マルチホームホスト 複数の物理インタフェースをもち、バケット転送を行わないシステム。マルチホームホストでは軌道線制御プロトコルを実行できます。

メーター 特定クラスのトラフィックフローの速度を測定する diffserv アーキテクチャのモジュール。IPQoS 実装には、token mt および tsaw tclmt という 2 つのメーターがありま

メッセージ認証コード (MAC) データの整合性を保護し、データの出所を明らかにするコード。MAC は盗聴行為には対応できません。

ユーザーペリピア snoビルのマークを実装する 3 ビットの値。VLAN デバイスのネットワーク上で Ethernet データグラムが転送される方法を定義します。

ユーザーサービスクラスアドレス IPv6 が有効なノードの単一インタフェースを識別する IPv6 アドレス。ユーザーサービスクラスアドレスは、サイト接続辞や、サブネット ID、インタフェース ID などからなります。

リダイレクト 特定の端点に到達するために、ホストに対して最適な最初のホップノードを、ルーターが通知すること。
リンク - ローカル・アドレス

IPv6 では、自動アドレス設定などのために、単一リンク上でアドレスを指定するために使用することを表します。デフォルトでは、リンク - ローカル・アドレスはシステムのMACアドレスから作成されます。

リンク層 IPv4/IPV6 のすぐ下の層。

ルーター 複数のインタフェースを通常もち、経路制御プロトコルを実行し、パケットを転送するシステム。システムが PPP リンクのエンドポイントである場合は、ルーターとしてのインタフェースを１つだけ持つようなシステムを構成できます。

ルーター広告 ルーターが、各種のリンクパラメータおよびインターネットパラメータと共に、その存在を定期的あるいはルーター要請メッセージに応じて通知すること。

ルーター発見 ホストが、接続されているリンク上にあるルーターを特定すること。

ルーター要請 ホストがルーターに対し、次に予定されている時刻ではなく、ただちにルーター広告メッセージを送信するために要請すること。

ローカル使用アドレス ローカルの経路制御可能な範囲だけを対象とするユニキャストアドレス (サブネット内またはネットワーク内)。また、ローカルまたはグローバルな一意の範囲を対象とすることもできます。
索引

A
ATM, IPMP サポート, 305

B
BSSID, 「WiFi」を参照

C
CPU pool プロパティー, 414
CPU の割り当て, 418–419
CPU プールリソース, リンクへの割り当て, 416

D
dladm コマンド
VLAN の構成, 258–261
WiFi の構成, 213
集約の変更, 249
データリンク
MTU サイズの変更, 165–167
情報の表示, 162–163
データリンクの削除, 163
名前の変更, 160
物理属性の表示, 161
ネットワークリソースの管理, 397
dlstat コマンド, 425, 429
show-phys, 432–433

E
ESSID, 「WiFi」を参照
/etc/default/mpathd ファイル
「IPMP、構成ファイル」を参照

F
FAILBACK=no モード, 288
flowadm コマンド, 419–424
フロー上のリソースの管理, 397
flowstat コマンド, 425

I
ifconfig コマンド
STREAMS モジュールの順番チェック, 305
とipadm コマンド, 204
in.mpathd デーモン, 「IPMP, in.mpathd デーモン」を参照
ip-nospoof, リンク保護タイプ, 390
ipadm
set-addrprop, 188
show-addrprop, 188
ipadm コマンド
IPMP インタフェースの作成, 309–310
IPMP 用のサブコマンド, 309
IP アドレスのプロパティーの設定, 188
IP インタフェースの構成, 183
TCP/IP プロパティーの管理, 179
インタフェースのplumb, 184
索引

ipadm コマンド (続き)
インタフェースの監視，198
インタフェースの削除，247
と ifconfig コマンド，204

IPMP
ATM のサポート，305
Ethernet のサポート，305
in.mpathd デーモン，280,285
ipmpstat を使用した情報の表示，322–330
IP の要件，283
インタフェース構成のタイプ，281
インタフェースの交換，DR，320–322
概要，270–271
管理，312–316
基本的な要件，304–306
検査用アドレス，282
構成ファイル，280,319–320
修復検出，287–288
障害検出，284,293
ソフトウェアコンポーネント，280
ターゲットシステム，構成，317–318
データアドレス，282,291
動的再構成，288–290,292
トークンリングのサポート，305
匿名グループ，287,293
とリンク集約，272–273
負荷分散，271,295
ブロードキャスト，297
ブロードキャストフィック，284–286
用語，291
ipmpstat コマンド，269–270,281,301,322–330
IPMP インタフェース，269–270,294
IPMP グループの構成，309–310
情報の表示，274,322–330
ベースとなるインタフェースの故障，274
IPMP グループ，294
「IPMP インタフェース」も参照
DHCP を使用した構成，306–309
NIC の交換，DR 経由，290
NIC の取り外し，DR 経由，290
新しい NIC の接続，DR 経由，289
アドレスの追加または削除，314–315
インタフェースのグループからの削除，313
インタフェースのグループへの追加，312–313

IPMP グループ (続き)
グループ間でのインタフェースの移動，315
グループ障害，286
計画タスク，304–306
情報の表示，322–330
IP アドレス，プロパティ，188
IP マルチバス (IPMP)，「IPMP」を参照

L
LACP (Link Aggregation Control Protocol)
変更 LACP モード，249
モード，245
LLDP，331
Oracle Solaris 内のコンポーネント，331–332
TLV ユニット，335–337
エージェント，333–337
動作モード，333–337
LLDPU，「LLDP、TLV ユニット」を参照
LLDP エージェント，「LLDP、エージェント」を参照

M
mac-nospoof，リンク保護タイプ，390
MAC アドレス
IPMP のための要件，304–306
一意であることが確認，181–183
MAC クライアント，399
構成，401
ソフトウェアベース，399,405
ハードウェアベース，399,401
リングの割り当て，401
MIB，333–337
MTU，「最大伝送単位」を参照

N
/net/if_types.h ファイル，305
netstat コマンド，WiFi リンク上のパケットフローの確認，218
索引

R
RCM (Reconfiguration Coordination Manager) フレームワーク, 290
restricted, リンク保護タイプ, 390

S
STREAMS モジュール, とデータリンク, 176

T
TCP/IP パラメータ, ipadm コマンドでの設定, 179
TLV, 「LLDP, TLV ユニット」を参照
Trunking, 「集約」を参照

V
VLAN
PPA ハッキング, 256
VLAN 名, 256
計画, 257–258
構成, 253–267
サンプルシナリオ, 253
定義, 253–267
トポロジ, 254–256
物理接続点 (PPA), 256
リンク集約上での作成, 261–262
VNIC
CPU プールリソースの割り当て, 416
plumb, 381–384

W
WEP キーの構成, 219
WiFi
IEEE 802.11 仕様, 210
WEP キーの生成, 219
WiFi ネットワークのタイプ, 210
WiFi ネットワークへの接続, 212, 214, 215
WiFi の構成例, 215
WiFi を実行するためのシステムの準備, 212
WiFi (続き)
暗号化された通信の例, 220
拡張サービスセット識別子 (ESSID), 213
基本サービスセット識別子 (BSSID), 213
サポートされているインタフェース, 212
セキュリティー保護された WiFi リンク, 218
接続の暗号化, 219
定義, 210
ホットスポット, 210
リンクの監視, 217
例, リンク速度の設定, 218

あ
アクセスポイント, WiFi, 210, 212
アクティブ・アクティブインタフェース
IPMP, 309–310, 310–312
アクティブ・アクティブインタフェース
IPMP, 281
アクティブ・スタンバイインタフェース
IPMP, 281
アドレスの移行, 270
「IPMP, データアドレス」も参照

い
インタフェース
IPMP での構成タイプ, 281
IPMP での修復検出, 287–288
MAC アドレスが一致であることの確認, 181–183
VLAN, 253–267
WiFi のタイプ, 212
インタフェースでの STREAMS モジュールの順番, 305
永続的な構成の作成, 186
構成
VLAN の一部として, 258–261
WiFi インタフェース, 213
集約, 246–248
データリンク上, 184
スタンバイ, IPMP, 281
インタフェース専用の CPU, 418–419
索引

インタフェースの監視、ipadmコマンドの使用，198

え
永続的なリンク構成，作成，186

か
仮想化とサービスの品質，395

く
グループ障害，IPMP，286

け
検査用アドレス
「IPMP、検査用アドレス」を参照

こ
構成，リンク保護，391–393

さ
最大伝送単位 (MTU)，165–167

し
ジャンボフレーム，サポートの有効化，165–167
修復検出時間，287–288
集約
機能，241
作成，246–248
定義，241
トポロジ
基本，242

す
推移的ブローブ，284–285
スイッチの構成
LACP (Link Aggregation Control Protocol)
モード，245, 249
集約トポロジ，242
スタンバイインタフェース
「ifconfigコマンド、IPMP用のオプション」も参照
IPMP グループでの役割，281

せ
静的アドレスのグループ化，「リングのグループ化」を参照
セキュリティの注意点，WiFi，218

た
ターゲットシステム、IPMP，手動構成，317–318
ターゲットのブローブ，IPMP，280

て
データアドレス，「IPMP、データアドレス」を参
照
データリンク
「dladm コマンド」も参照
Ethernet パラメータ, 168–170
MTU サイズ, 165–167
STREAMS モジュール, 176
カスタマイズされた名前の使用規則, 30–31
情報の表示, 162–163
データリンクの削除, 163
命名規則, 26–31
リンク上での IP インタフェースの構成, 184
リンク速度パラメータ, 167–168
リンクの名前の変更, 160
リングプロパティの管理, 158
リング名, 26–31
IPMP 構成での使用, 273–274

と
動的再構成 (DR)
「ネットワークインタフェースカード (NIC)」も参照
IPMP との相互運用, 288–290
IPMP との連携, 320–322
NIC の交換, 173
インタフェースの操作, IPMP, 289, 290,
320–322
カスタマイズリング名による柔軟性, 31
定義, 292
動的なリングのグループ化、「リングのグループ化」を参照
トークンリング, IPMP サポート, 305
匿名グループ, 287, 293
特権ポート, ipadm コマンドでの設定, 194

な
なりすまし, リンクの保護, 389–391

ね
ネットワークインタフェースカード (NIC)
Ethernet パラメータ設定, 168–170
ネットワークインタフェースカード (NIC) (続き)
NIC ドライバーの公開および非公開プロバ
ティ, 164
交換, DR の使用, 173, 320–322
交換, DR を使用, 290
障害とファイルオーバー, 293
動的再構成, 292
リンク速度パラメータ, 167–168
ネットワーク構成プロファイル (NCP), 153–154
ネットワーク使用の監視, 425
ネットワークスタック, 22, 24
ネットワークトラフィックの統計情報, リングご
との, 432–433
ネットワークの統計情報, 「ネットワーク使用の
監視」を参照
ネットワークリソースの管理, 395
実装のための dladm コマンド, 397
フローの使用, 396
リンク, 395
ネットワークレーン, 395
ソフトウェアレーン, 425
ハードウェアレーン, 425

は
ハードウェアベースのクライアント, 399
ハードウェアリング, 399–414

ふ
負荷分散, 271, 295
集約間, 244
物理インタフェース, 242–243
「インタフェース」も参照
物理接続点 (PPA), 256
フロー, 396, 419–424
フロー制御, 「フロー」を参照
プローブデータ、IPMP, 定義, 297
プローブトラフィック, 284–286
プローブベースの障害検出, 284–286
「IPMP、検査用アドレス」も参照
「IPMP、検査用アドレスを使用しない」も参
照
索引

プローブベースの障害検出 (続き)
推移的プローブ, 284–285
ターゲットシステムの構成, 316–320
と検査用アドレス, 285–286

ヘ
ベースとなるインタフェース, 298

ほ
ホットスポット, WiFi
定義, 210
ホットスポットを探す, 211
ポリシー, 負荷分散, 244

む
無線インタフェース, 「WiFi」を参照

ゆ
ユーザー指定の名前, 「データリンク、リンク名」を参照

り
リソース制御, 「ネットワークリソースの管理」を参照
リンク集約, 「集約」を参照
リング、送信と受信, 399–414
リングのグループ化
「リング割り当て」も参照
動的および静的, 399–414
リンクベースの障害検出, 286–287
リンク保護, 389–391
構成, 391–393
リンク保護タイプ, 390–391
ip-nospoof, 390
mac-nospoof, 390

Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化・2011年12月