パケットづくりではじめる ネットワーク入門

第24回 標準的パケット送受信ライブラリlibpcapのWindows版WinPcap

坂井 弘亮

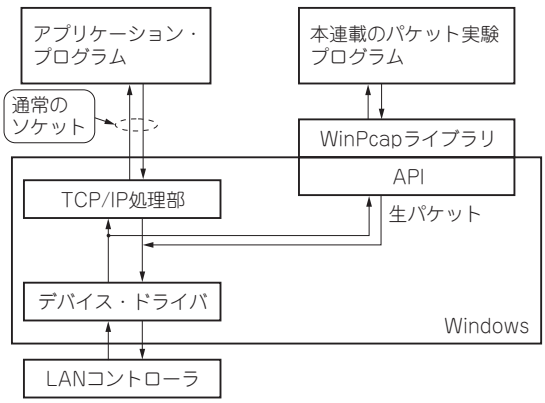

図1　標準的パケット送受信ライブラリ「libpcap」のWindows版「WinPcap」

＊BSDのBPFやLinuxのRAWソケットを直接使う代わりに，libpcapというライブラリを利用すれば，パケットの送受信をプラットフォームOSに依存せずに行うことができます．WindowsではlibpcapのWindows版ともいえる，WinPcapというライブラリが利用できます(**図1**)．

今回は，WinPcapを用いてパケット・ライブラリをWindows対応にします．「ping応答ツール」をWindows向けにビルドして，パケット・ライブラリのWindows上での動作を確認します(「ping応答ツール」の詳細は2015年12月号の第5回参照)．

BSDでもLinuxでも使える標準的パケット送受信ライブラリlibpcapのWindows版WinPcap

● 特徴

WinPcapはパケットの直接送受信を行うための代表的なライブラリであるlibpcapを，Windowsに移植したものです．ネットワーク・アナライザであるWiresharkをWindowsにインストールする際に，WinPcapも追加でインストールされます．

パケットを扱うプログラムがlibpcapを利用して書かれている場合，パケットの送受信などのlibpcapがサポートしている部分に関しては，比較的簡単にWindowsに移植できます．

しかし実際には考慮が必要な部分もあり，そのままビルドし直すだけというわけにはいきません．

● 使うときの注意点

WinPcapを利用する際の注意点を説明します．

▶その1：インターフェース名の指定方法がlibpcapと違う

libpcapでは，送受信のインターフェースを`pcap_open_live()`によってオープンする際に，インターフェース名を指定します．インターフェース名は，FreeBSDではem0やrl0，Linuxではeth0やeth1といった文字列です．ping応答ツールでは，動作するインターフェースを，実行時のコマンドライン引き数上でインターフェース名によって指定しています．

しかしWindowsでは，インターフェース名は16進文字列を含む長い文字列で表現されるため，コマンドラインなどから指定することは不便です．このため，まずはインターフェース一覧を取得し，その中からインターフェースを指定するようにします．

インターフェース一覧は，`pcap_findalldevs()`というライブラリ関数によって取得できます．`pcap_findalldevs()`によってインターフェース一覧を取得するサンプルを**リスト1**に示します．

`pcap_findalldevs()`を呼び出すことで，インターフェースの一覧情報がリンク・リストの構造で返されるので，リンク・リストを追うことで一覧を出力しています．**リスト1**を改造すれば，例えばインターフェースをリンク・リストの先頭から何番目かで指定して，その名前を得ることができます．

リスト2は，引き数として与えられた番号のインターフェースを検索し，その名前を返す関数の例です．このようにして得られたインターフェース名を`pcap_open_live()`に渡せば，そのインターフェースを送受信用にオープンすることができます．

▶その2：利用できないライブラリ関数がある

libpcapでは利用できるが，WinPcapでは利用できないライブラリ関数が幾つかあるようです．

本連載のプログラムのソースコードは以下の筆者のホームページからダウンロードすることができます．ライセンスはKL-01というもので，組込み機器などでも自由に利用することができます．
http://kozos.jp/books/interface/ethernet2/