第2章 IPv6アドレスの管理からパケット送受信処理まで

ステップ①… IPv6通信の実装

柚山 大哉

第2章では，ルータ開発を行う上で欠かせない，IPv6通信に関する機能を実装します．

ここでは，IPv6アドレスを管理する構造と，IPv6パケットを受信したときの処理，送信するときの処理に焦点を当てて説明します．

IPv6アドレスは構造体で管理

● Linuxで使っている構造体を流用する

IPv4の場合はアドレスが32ビットだったのでint型で管理できましたが，IPv6の場合は128ビットあるので構造体（複数の変数を束ねたもの）で管理します．今回は，Linuxのヘッダにあるものをincludeして，そのまま使うことにします［**リスト1**(**a**)］．

▶流用するメリット

Linuxの構造体をそのまま使うことで，IPv6アドレスの文字列化や，文字列からデータに変換するなどのLinux関数をそのまま利用できるメリットがあります．本特集の趣旨としては，自作すべきところですが，構造体自体は特に機能を持たないことや，Linuxそのものを使うことで大幅に便利になることから，今回は流用することにしました．

▶IPv6アドレス構造体の中身

LinuxのIPv6構造体であるin6_addrを**リスト1**(**b**)に示します．

in6_addr構造体は，union型を使って16バイト（128ビット）のデータに，1バイトごとや，2バイトの情報8つ分，4バイトの情報4つ分など，異なる粒度でアクセスできるようになっています．s6_addr32などのマクロが定義されているので，uint8_tの配列としてアクセスするときはs6_addr32，uint16_tの配列としてアクセスするならs6_addr16，uint32_tの場合はs6_addr32を使います．

● インターフェースとIPv6アドレスのひも付け

ルータのネットワーク・インターフェースとIPv6アドレスの情報を持つために，ipv6_deviceという構造体を定義して使用することにします（**リスト2**）．

ipv6_deviceでは，IPv6アドレスとプレフィックスの他にスコープを保持します．IPv6では，1つのインターフェースで複数のアドレスを持つことがよく行われているため，スコープでアドレスを使用する優先度を決定します．

リスト1　IPv6ヘッダを管理する構造体はLinuxで利用されているものを流用する

IPv6アドレスの文字列化や，文字列からデータに変換するなどのLinux関数をそのまま利用できるメリットがある

```
 struct ipv6hdr {
#if defined(__LITTLE_ENDIAN_BITFIELD)
    __u8            priority:4,
                    version:4;
#elif defined(__BIG_ENDIAN_BITFIELD)
    __u8            version:4,
                    priority:4;
#else
#error  "Please fix <asm/byteorder.h>"
#endif
    __u8            flow_lbl[3];

    __be16          payload_len;
    __u8            nexthdr;
    __u8            hop_limit;

    __struct_group(/* no tag */, addrs,
                                        /* no attrs */,
        struct  in6_addr    saddr;
        struct  in6_addr    daddr;
    );
};
```

(**a**)(1) IPヘッダ構造体ipv6hdr

```
struct in6_addr {
    union {
        __u8                u6_addr8[16];
#if __UAPI_DEF_IN6_ADDR_ALT
        __be16              u6_addr16[8];
        __be32              u6_addr32[4];
#endif
    } in6_u;
#define s6_addr             in6_u.u6_addr8
#if __UAPI_DEF_IN6_ADDR_ALT
#define s6_addr16           in6_u.u6_addr16
#define s6_addr32           in6_u.u6_addr32
#endif
};
```

(**b**)(2) IPv6アドレス構造体in6_addr