

Student Research Seminar Ws2008

desproute2 - Eine Layer L2.5 Routing Library

Philipp Schmidt

Department of Mathematics and Computer Science
Institute of Computer Science
25. November, 2008

1. Das wichtigste zuerst....
2. Die Ideen
3. Designprinzipien
4. Ein einfacher Routingdaemon

...das Acronym.

...das Acronym.

Library for the **DES**-Testbet by **Phils** to
implement **Routing** an layer **2**

Mesh-Routing für IP birgt einige Probleme

- IP, ARP & Co sind nicht erweiterbar...
...viele simulierte Mesh-Protokolle gehen aber davon aus sie könnten IP oder ARP erweitern
- Reaktives Routing durchbricht viele Annahmen die zur IP-Implementierung gemacht wurden...
...unzählige Unklarheiten am Rande

Kernel-Implementierungen von Routingprotokollen sind komplex

- Der Linux-Kernel geht nur von klassischem, proaktivem Routing aus...
... dafür genügt es Routen in die Kernel-Routingtabelle einzutragen.
- Den IP-Stack für reaktives Routing zu verändern ist sehr komplex...
...den Paketfilter dazu zu mißbrauchen mindestens unschön und immernoch komplex.
- Kernel-Programmierung ist fehlerträchtig...
...und schwer zu debuggen.

Die Lösung:

- Eine eigene Schicht für das Mesh-Routing wird eingeschoben (vergleichbar zu MPLS oder ATM in TK-Netzen).
- Auf den Kommunikationsendpunkten stellt ein virtuelles Netzwerkinterface die Verbindung zwischen Mesh-Routing und IP-Stack dar.
- Implementierung erfolgt komplett im Userspace (eine Kernelportierung ist möglich - aber nicht geplant).

- *Einfacher Kontrollfluss* - für jedes empfangene Paket ruft die Library registrierte Callbacks.
- *Einfacher Paketbau/Analyse* - für die Implementierung des Protokolls soll ich nicht über die Bitschiebereien nachdenken müssen.
- *Einfach zu Debuggen* - komplette Implementierung im Userspace mit vielen Loggin-Möglichkeiten.


```
int despr2_msg_getext(const despr2_msg_t* msg, despr2_ext_t** ext,
                    uint8_t type, int index) {
    int i = 0;
    despr2_ext_t *exti;
    if(ext != NULL)
        *ext = NULL;
    exti = (despr2_ext_t *)(((uint8_t *) msg + DESPR2_MSGLEN);
    while( (uint8_t *)exti <
           ( (uint8_t *) msg + (size_t) ntohs(msg->hlen) ) ) {
        if ( type == exti->type || type == DESPR2_EXT_ANY) {
            if ( i == index && ext != NULL)
                *ext = exti;
            i++;
        }
        exti = (despr2_ext_t *)(((uint8_t *) exti) + (size_t) exti->len);
    }
    if ( i <= index )
        i = -i;
    return(i);
}
```

Demo...

Danke!