

# クロスレイヤ資源情報取得・発見 プロトコルおよび API の実装

Katsushi Kobayashi  
理研・AICS



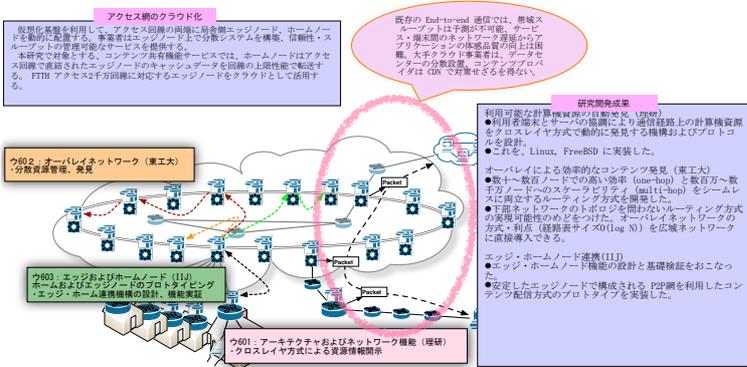
RIKEN Advanced Institute for Computational Science

Copyright (C) 2012, Katsushi Kobayashi. All rights reserved.

## DYPOP プロジェクト

平成23年度「大規模コンテンツ配信基盤を実現するアクセス網のクラウド化」  
の研究開発目標・成果と今後の研究計画

1. 実施機関・研究開発期間・研究開発費  
● 独立行政法人理化学研究所(幹事者)、国立大学法人東京工業大学、株式会社インターネットイニシアティブ
2. 研究開発の目標(27年6月)  
アクセス回線の両端のエッジ・ホームノードで構成されるアクセス網クラウドシステムに必要な技術開発をおこなう。コンテンツ配信アプリケーションをクラウド上で実装・展開、アクセス網クラウドの有用性を示す。
3. 研究開発の成果



- アクセス網にコンテナデータセンター(DC)を直結
- DCのサーバ資源を仮想化でサービス事業者に分割提供
- サービスをアクセス網の対向側で動作させ低遅延とアクセス網帯域を有効活用
- US Ignite も同じ考え方
- 宅側機器を DC で稼働させることでアクセス回線+宅側機器の電力削減

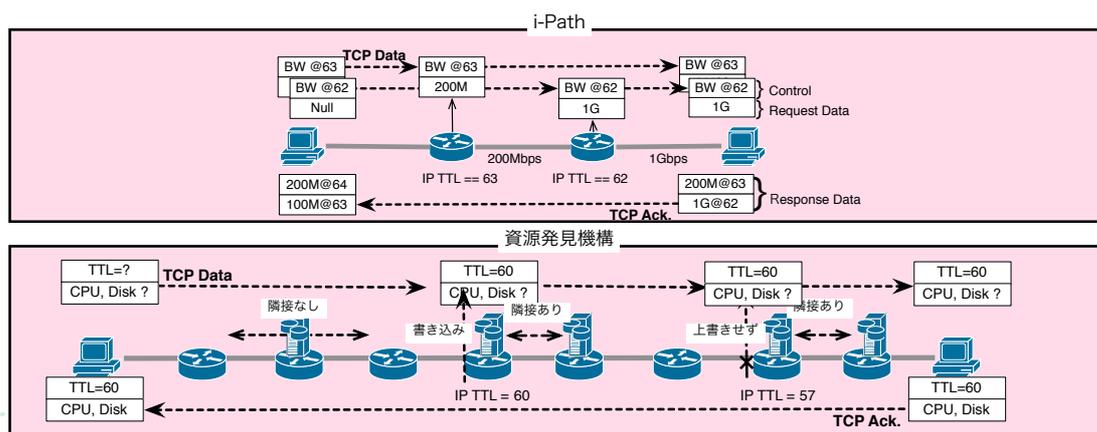


RIKEN Advanced Institute for Computational Science



# クロスレイヤによる資源情報開示

- ネットワーク機能：i-Path 拡張による資源発見機構
  - i-Path のネットワーク情報把握には経路ルータ数分のパケットが必要
  - 端末から送信されたi-Path の資源情報の照会パケットに対して、IP ヘッダ中の TTL 値(あるいはホップカウント)を shim ヘッダに書き込む、送信先から TTL 値を返送、1 パケット往復で資源の有無を把握



## クロスレイヤ資源発見機構の設計

- プロトコル
  - 資源情報要求、開示は IP option で提供
    - トランSPORTは既存のものが利用できる
    - 途中経路が未対応でも通信可能
- ルータ機能
  - 要求の判断と応答、資源情報設定 API
    - 利用可能な資源量は変化する
- ホスト機能
  - 要求内容の設定、開示情報の取得 API
  - 要求資源情報、開示された資源はソケット毎に管理
    - ソケット API による実装

# クロスレイヤ資源発見機構の実装にあたって

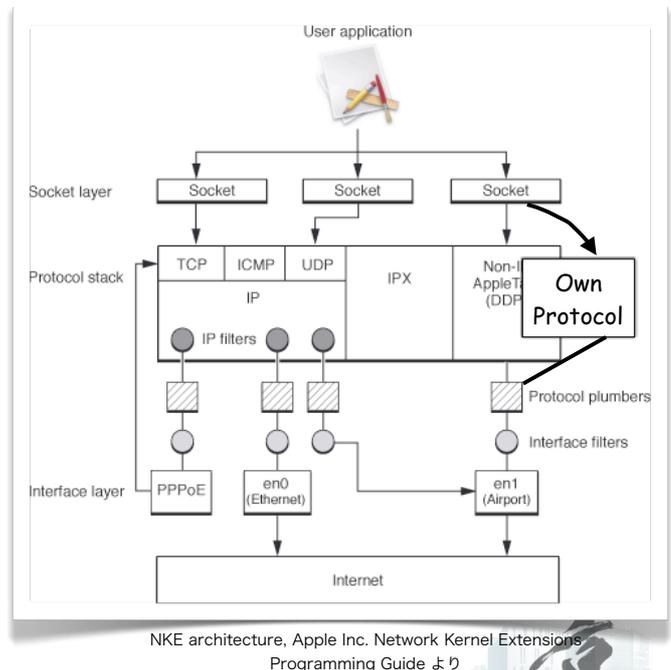
- 実装、その評価を通じてその有用性を示す
  - プロトコルの設計、実装はもはや当たり前
  - 実ネットワーク or テストベッドでのデモが不可欠
    - デモ対象は学術関係者、専門家ばかりではない
    - 利用シナリオを想定したわかりやすい内容
- ▶性能+見栄えに加えて、Commodity OS + モバイルデバイス
  - ノード機能+ホスト側機能

## FW 支援フレームワークを利用したプロトコル実装

- オリジナル配布 kernel の置き換えは困難
  - ソース開示、セキュリティパッチ、動作検証....
- Userland の実装は API が複雑に...
  - App. は kernel + userland API の対応が求められる
- Loadable Kernel module + FW 支援フレームワーク
  - Linux (incl. Android) : netfilter/iptables
  - \*BSD: Packetfilter (pf)
  - OSX: Network Kernel Extension (NKE)
  - Windows: Windows Filtering Platform (WFP)
- プロトコルスタックの実装も視野に

# FW 支援フレームワーク contd.

- FW に必要とされる機能
  - interface, IP, Socket それぞれのレベルで data-plane を横取り
  - Socket API の追加、横取り
  - プロトコル状態遷移イベントの wrapper サポート
- プロトコルスタックの全部、一部を実装が可能に
  - NKE を利用した SCTP の実装も提供されている



NKE architecture, Apple Inc. Network Kernel Extensions Programming Guide より

## Interface/IP filter Support

```

struct SRIFEntry {
    TAILQ_ENTRY(SRIFEntry)    srif_list;
    struct iff_filter         srif_filter;
};
...
struct SRIFEntry srif[IFMAX];
for(i = 0; i < ifcount; i++){
    srif[i].srif_filter.iff_name = MYBUNDLEID;
    srif[i].srif_filter.iff_protocol = 0;
    srif[i].srif_filter.iff_input = sr_iff_in_fn;
    srif[i].srif_filter.iff_output = sr_iff_out_fn;
    srif[i].srif_filter.iff_event = NULL;
    srif[i].srif_filter.iff_ioctl = sr_iff_ioctl_fn;
    srif[i].srif_filter.iff_detached = sr_iff_detached_fn;
    TAILQ_INSERT_TAIL(&sr_iflist, &srif[i], srif_list);
    ret = iflt_attach(ifnetp[i], &(srif[i].srif_filter),
        &(gsr_if_filter[i]));
}
    
```

```

errno_t sr_iff_in_fn(void *cookie, ifnet_t interface,
    protocol_family_t protocol, mbuf_t *data, char **frame_ptr)
{
    .....
    /* If stop packet processing */
    return EJUSTRETURN;
    /* If continue to normal path */
    return 0;
}
    
```

```

errno_t sr_iff_out_fn(void *cookie, ifnet_t interface,
    protocol_family_t protocol, mbuf_t *data)
{
    .....
    /* If stop packet processing */
    return EJUSTRETURN;
    /* If continue to normal path */
    return 0;
}
    
```

```

struct ipf_filter sr_ipf_filter = {
    &sr_enable,          /* void * cookie */
    MYBUNDLEID,         /* char * filter name */
    sr_ipf_input,       /* ipf_input_func */
    sr_ipf_output,      /* ipf_output_func */
    sr_ipf_detach_func  /* ipf_detach_func */
};
    
```

```

error_t sr_ipf_input(void *cookie, mbuf_t *data, int offset,
    u_int8_t protocol)
{
    .....
    /* If stop packet processing */
    return EJUSTRETURN;
    /* If continue to normal path */
    return 0;
}
    
```

```

error_t sr_ipf_output(void *cookie, mbuf_t *data,
    ipf_pktopts_t options)
{
    .....
    /* If stop packet processing */
    return EJUSTRETURN;
    /* If continue to normal path */
    return 0;
}
    
```

# Socket filter Support

```
struct sflt_filter sr_sflt_ip4_tcp = {
    SIRENS_HANDLE4TCP, /* sflt_handle */
    SFLT_GLOBAL | SFLT_EXTENDED, /* sf_flags */
    MYBUNDLEID, /* sf_name */
    sr_unregister_fn, /* sf_unregister_func */
    sr_attach_fn, /* sf_attach_func */
    sr_detach_fn, /* sf_detach_func */
    NULL, /* sf_notify_func */
    NULL, /* sf_getpeername_func */
    NULL, /* sf_getsockname_func */
    NULL, /* sf_data_in_func */
    NULL, /* sf_data_out_func */
    sr_connect_in_fn, /* sf_connect_in_func */
    sr_connect_out_fn, /* sf_connect_out_func */
    sr_bind_fn, /* sf_bind_func */
    sr_setopt_fn, /* sf_setopt_func */
    sr_getoption_fn, /* sf_getoption_func */
    NULL, /* sf_listen_func */
    NULL, /* sf_ioctl_func */
    /* sr_accept_fn is in extension part in filter */
};
```

```
errno_t sr_connect_out_fn(void *cookie,
    socket_t so, const struct sockaddr *to)
{
    /* Check and mark protocol support */
    ....
    return 0;
}
```

```
errno_t sr_connect_in_fn(void *cookie,
    socket_t so, const struct sockaddr *to)
{
    /* Check and mark protocol support */
    ....
    return 0;
}
```

```
errno_t
sr_accept_fn(void *cookie, socket_t so_listen, socket_t so,
    struct sockaddr *local, struct sockaddr *remote)
{
    /* Check and mark protocol support */
    ....
    return 0;
}
```

```
errno_t sr_bind_fn(void *cookie,
    socket_t so, const struct sockaddr *to){
    /* Check and mark protocol support */
    ....
    return 0;
}
```

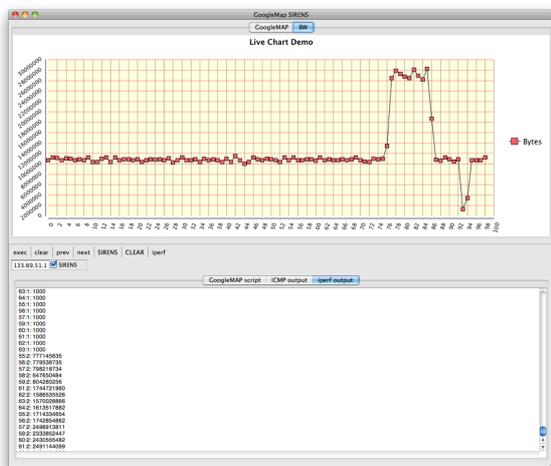
```
errno_t sr_getoption_fn(void *cookie,
    socket_t so, sockopt_t opt)
{
    int error = 0;
    switch(sockopt_name(opt)){
        case IPSIRENS_ABC: /* New socket option */
            ....
            error = 0;
    }
    return error;
}
```

```
errno_t sr_setopt_fn(void *cookie,
    socket_t so, sockopt_t opt)
{
    int error = 0;
    switch(sockopt_name(opt)){
        case IPSIRENS_ABC: /* New socket option */
            ....
            error = 0;
    }
    return error;
}
```

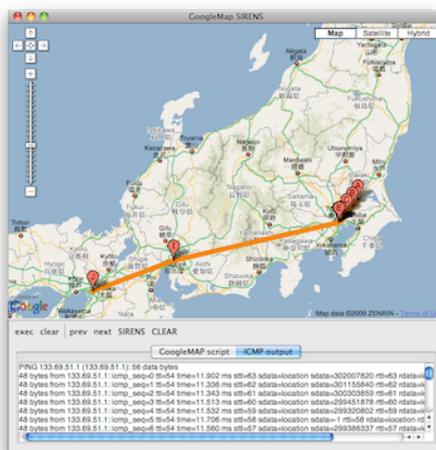


## アプリケーションの例

- 以下の例は Java SWT, BIRT, Android + クロスレイヤプロトコルのアプリケーション例



経路上の利用帯情報域の実時間収集



経路上のルータ位置情報をオーバーレイした例



Android の実装  
早稲田大学、持永氏より



# まとめ

- **FW** 支援フレームワークを用いた“新世代”ネットワーク  
プロトコルの実装、アプリケーション実装

- 本発表の一部は、独立行政法人情報通信研究機構、高度通信・放送研究開発委託研究の成果である
- 早稲田大学・後藤、MRI 持永、村瀬、東工大・首藤、IIJ 石田、山本、曾我部らに感謝します（敬称略）